Maximilán Strémy

Combined Discrete Control Systems

Scientific Monographs in Automation and Computer Science

Edited by
Prof. Dr. Peter Husar (Ilmenau University of Technology) and
Dr. Kvetoslava Resetova (Slovak University of Technology in Bratislava)

Vol. 2

COMBINED DISCRETE CONTROL SYSTEMS

Maximilán Strémy

Universitätsverlag Ilmenau
2012

Impressum

Bibliographic information of the German National Library
The German National Library lists this publication in the German national bibliography, with detailed bibliographic information on the Internet at http://dnb.d-nb.de.

Author's acknowledgement to Emília Mironovová for translation.

This scientific monograph originated from the author's dissertation thesis defended at the Slovak University of Technology in Bratislava, Faculty of Materials Science and Technology in Trnava.

Reviewers:
Boris Roháľ-Iľkiv, Professor, Ph.D.
Juraj Miček, Assoc. Professor, Ph.D.
Augustín Gese, Ph.D.

Author's contact address:
Maximilián Strémy, Ph.D.
Slovak University of Technology in Bratislava
Faculty of Materials Science and Technology in Trnava

Ilmenau Technical University / University Library
Universitätsverlag Ilmenau
Postfach 10 05 65
98684 Ilmenau
www.tu-ilmenau.de/universitaetsverlag

Production and delivery
Verlagshaus Monsenstein und Vannerdat OHG
Am Hawerkamp 31
48155 Münster
www.mv-verlag.de

ISSN 2193-6439 (Print)
ISBN 978-3-86360-014-3 (Print)
URN urn:nbn:de:gbv:ilm1-2012100023

Titelfoto: photocase.com

Abstract

Combined control dynamic systems comprise both parts of discrete systems: real-time activated discrete dynamic systems as well as discrete event dynamic systems. The combination of cyclically recurring processes with stochastically generated events significantly intervenes into the control, while also affecting the choice of the right sample period, as it is necessary to process all the generated interruptions within the cycle. The current monograph focuses on the analysis of combined control systems and determination of the stochastic event element of the sample period. It proposes an appropriate sample period for combined control systems and introduces a methodology for determining the impact of a stochastic element on the system stability and properties. The main contribution of the monograph is the design of an adaptive algorithm for events processing and auto-correction of the event element of sample period.

Key words

adaptive algorithm, combined control systems, probability, sample period, stochastic system, events processing.

LIST OF ABBREVIATIONS

A/D (A/Č)	-	Analog-Digital
COM	-	Component Object Model
D/A (Č/A)	-	Digital-Analog
DB	-	Database
DCOM	-	Distributed Component Object Model
DCS	-	Distributed Control System
DEDS (DUDS)	-	Discrete event dynamic system
DS	-	Dynamic System
EDD	-	Earliest Due Date First
EDS (UDS)	-	Event Dynamic System
ERD	-	Earliest Release First
ERP	-	Enterprise Resource Planning
FCFS	-	First-Come-First-Serve
HTTP	-	Hypertext Transfer Protocol
ILAN	-	Industrial local area network
LAN	-	Local Area Network
LPT	-	Longest Processing Time
MES	-	Manufacturing Execution System
MPI	-	Multi-Point Interface
OB	-	Organization Block
ODBC	-	Open Database Connectivity
OLE	-	Object Linking and Embedding
OPC	-	OLE for Process Control

PC	-	Personal Computer
PLC	-	Programmable Logic Controller
RR	-	Round Robin
SCADA/HMI	-	Supervisory Control and Data Acquisition/Human Machine Interface
SIRO	-	Service In Random Order
SPT	-	Shortest Processing Time
SQL	-	Structured Query Language
SQND	-	Shortest Queue at the Next operation
TCP/IP	-	Transmission Control Protocol/Internet Protocol
WSPT	-	Weighted Shortest Processing Time

LIST OF SYMBOLS

T_0	- sample period
T_P	- event time constant of sample period
T_R	- time-driven part of sample period
T_{Pk}	- time necessary to process the kth event
τ	- time constant
T_{0var}	- maximum value of variable sample period
T_{cyc}	- period of cyclic processing in Siemens S7 stations
T_O	- sample period
T_R	- time-driven element of sample period
T_P	- event time constant of sample period
T_O	- sample period
T_R	- time-driven element of sample period
T_a	- mean time value at normal probability distribution
T_c	- time necessary to service all arisen interruptions
N	- number of interruptions in related period
μ	- arithmetic average of the time necessary to service the arisen events
σ	- mean quadratic deviation
$f(t)$	- density of probability distribution
$F(t)$	- distribution function
Δ	- determined time coefficient
T_A	- time necessary to perform all possible events in system
K_{min}	- minimum number of steps for distribution processing all events

$K(\tau)$	- autocorrelation function
$S_x(\omega)$	- power spectral density
$K(\tau)$	- autocorrelation function
$S_x(\omega)$	- power spectral density
D	- dispersion
σ	- mean quadratic error
N	- intensity of random process
$P_n(k)$	- Poisson's probability value at n-attempts and k-events
k	- number of events
n	- number of individual attempts
p	- examined value
q	- $q=1-p$
λ	- mean Poisson's parameter
x	- number of interruptions
T_A	- time necessary perform all possible events in system
T_O	- sample period
T_R	- time-controlled element of sample period
K_{min}	- minimum number of steps for the distribution processing of all events
t_u	- period of performing service programs of events
$S(t)$	- signal amplitude
v	- mean values of arising events
t	- time
$K(\tau)$	- autocorrelation function
$F(\omega)$	- spectral density of energy distribution
h	- amplitude level

T_O - sample period

T_R - time-driven element of sample period

T_P - event time constant of sample period

T_o - sample period

W(p) - transfer of continual system

W(z) - transfer of discrete system

ω - circular frequency

A - amplitude

N - maximum number of consecutive sample periods with the value exceeding the double of probability-determined event time component $2*T_{Podhad}$

Number - real number of periods with the value higher than $2*T_{Podhad}$

F_{r1} - No.1 queue of events

F_{r2} - No.2 queue of events

T_{Podhad} - probability-determined event time element of T_{REG} regulator sample period

T_R - constantly determined time element of sample period for performing cyclic instructions of time-controlled part of combined dynamic system

T_{REG} - sample period of programmable regulator

T_P - real time corresponding to the performance of program blocks and instructions of independent implemented events

τ_i - time of performing service routines of the i^{th} event

K - auxiliary variable for the differentiation of the implemented change of active queue

Change - auxiliary variable signalling the change of active queue.

INTRODUCTION INTO THE SUBJECT

One of the important aspects in dynamic systems assessment is their dependence on time or events. While dynamic systems of continuous constants are time-driven, discrete systems can be either time-driven or event-activated.

If status changes depend on the events occurring in discrete time moments, the systems can be referred to as discrete event dynamic systems (DEDS). In case status changes depend on the system periodic processing, monitoring and assessing in discrete time moments, the systems can be referred to as discrete time-driven systems.

In practice, control systems are based either on time-driven real-time systems or event systems, and they are usually encountered separately. Should both types of control systems be combined, (e.g. in major technological processes with a large number of sensors, actuators and the generation of events), each of the systems is generally conducted separately; one control unit is designed to work with the time-driven part, while the other one processes and responds to the events occurring in the system. The reason for separating both types of control systems may be either inadequate performance of management units, or lack of supportive methodologies and procedures for analysis and synthesis of such combined systems. Rapidly increasing the performance of the control units (either CPU or programmable logic controllers), along with more extensive distribution of systems and combinations of both types of systems (time-controlled systems with the event system in one system in real time), provide the space for maximum utilization of both types of control systems and their optimization, while minimizing related defects. The resulting fusion represents combined dynamic systems (CDS), also known as hybrid systems. Since the term “hybrid system” is also used in connection with the fusion of e.g. DCS and programmable logic systems, neural networks and fuzzy logic or a combination of electrical and mechanical drive units, a new concept has been introduced: the “Combined Dynamic System”.

A real management system usually includes both of the aforementioned management components: time and event. The complex dynamic properties of combined systems are defined by the priority of event discrete systems and time-driven systems. The structure of the primary control dynamic systems (along with their methods of synthesis and analysis, such as Petri nets, scheduling tasks, etc.) serves as the basis for the analysis and synthesis of the combined dynamic systems.

There are some methods suitable for the analysis and synthesis of time-driven or event systems, yet a generally accepted methodology for combined dynamic control systems enabling determination of the dynamic properties of such a system, including frequency spectrum of stochastically generated events or the optimum sample period regarding the ratio of time-driven and event elements in particular system, is missing.

The design of methodologies of analysis and synthesis of the combined dynamic system in Figure 1 represents selected procedures related to the analysis, synthesis and optimization of a single system’s parts, both separately (time-driven system, event system) and as a whole (combined systems). The methodology is based on the selected and generally known procedures for analysis and synthesis of discrete event systems and discrete real-time systems.

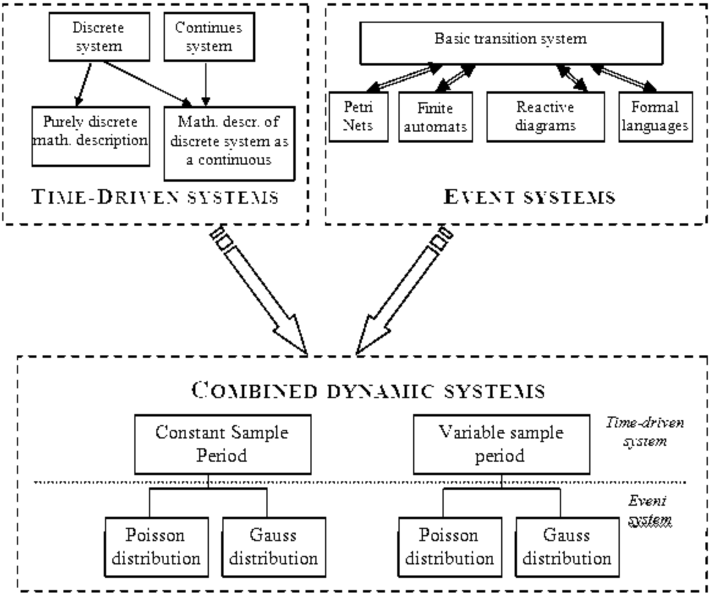


Fig. 1 *Designed methodology for analysis and synthesis of primary systems and related procedures determining time periods of combined dynamic systems*

The primary goals of the present monograph are as follows:

a) to select suitable tools and procedures for analysing both parts (time-driven system and event system) of a combined dynamic system,

b) to determine dynamic properties and quality parameters of combined dynamic systems,

c) to design a procedure for determining a sample period of combined dynamic systems in dependence from the distribution of emerging events corresponding to some of the probability laws in the system,

d) to design an adaptive algorithm determining a sample period and event processing in combined dynamic systems,

e) to analyse the influence of a stochastic element of the sample period in combined dynamic systems.

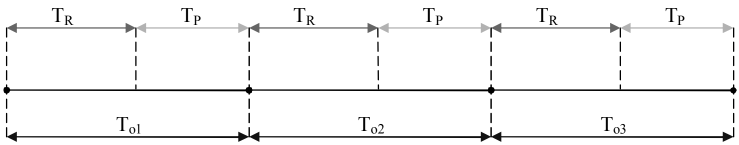
1. PRINCPLE AND CLASSIFICATION OF CDS

Combined control dynamic systems comprise both parts of discrete systems: discrete dynamic time-activated systems as well as discrete event systems. The combination of cyclically recurring processes in each processing period, with stochastically arising events intervening into the control, significantly influences the selection of a suitable sampling period or a cyclic processing, while processing all the necessary control instructions and service programs responding to emerging events, and simultaneously preventing their rejection, failure and other unacceptable states.

Regarding the sample period, combined dynamic systems can be classified into two categories:

a) those with a constant sample period (Fig. 2); when determining the period, it is necessary to determine also the ratio of the event element TP and the element of the time-driven system TR,

b) those with a changing sample period (Fig. 3); when it is necessary to determine the impact or relationship between the constant component (time-driven system) and the random component of the event (varying in dependence of the generated events).



$T_{const} = T_{o1} = T_{o2} = T_{o3}$

Fig. 2 *System with constant sample period T_0*

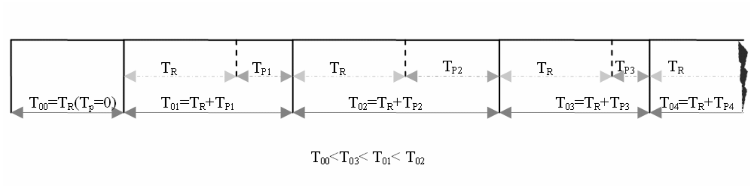
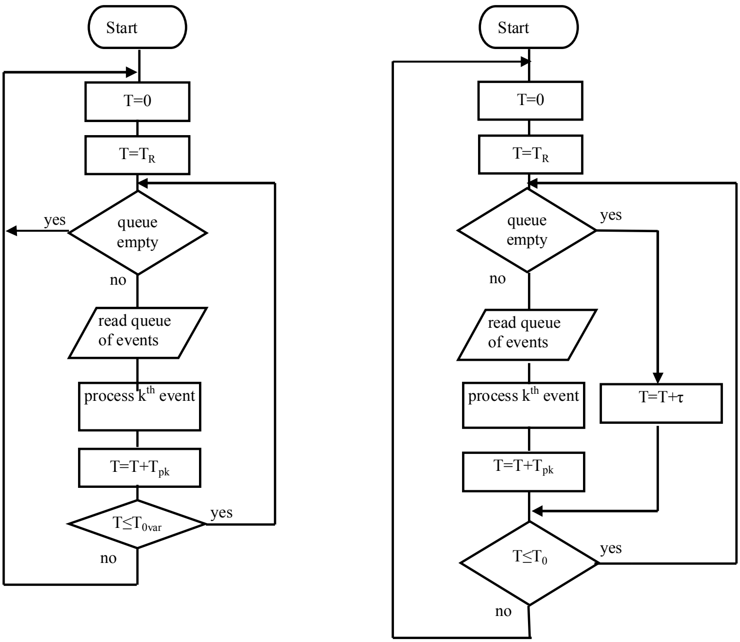


Fig. 3 *System with changing sample period* T_0

The considered implementation principle of both classified systems can be described by the algorithms expressed in the form of flowcharts (Fig. 4).



a) *b)*

Fig. 4 *Algorithms for combined systems - a)* T_{0var}*, b)* $T_0=const$

The event element of the sample period significantly influences the behaviour of the whole system, with its controllability, stability and the overall dynamic properties of combined systems determined by the ratio of event and time-driven discrete systems. While the time-driven part of combined discrete systems is carried out by means of constant cyclic monitoring, processing and assessing the system inputs and states, the event part in automated control systems is carried out by means of service interruptions related to the generation of some of the events. A special problem in such systems seems to be the method for determining the influence of stochastically changing event elements on their dynamic properties and control quality.

The following designed methodology (enabling the analysis of dynamic properties of combined dynamic systems and control quality as well as their synthesis) is relatively simple, yet precise. The design of the methodology considering both parts of the combined dynamic systems presented in the following chapters is based on the procedures for determining the optimum sample period by means of the statistic laws listed in Fig. 5. Besides the designed determination of the sample period using normal and Poisson's law of probability distribution, this monograph also describes the principle of the current automation control system implementation, spectral analysis of the energy distribution of stochastic systems, utilisation of white noise theory in combined dynamic systems, and the theory of density determination of probability of energy distribution of stochastic signal.

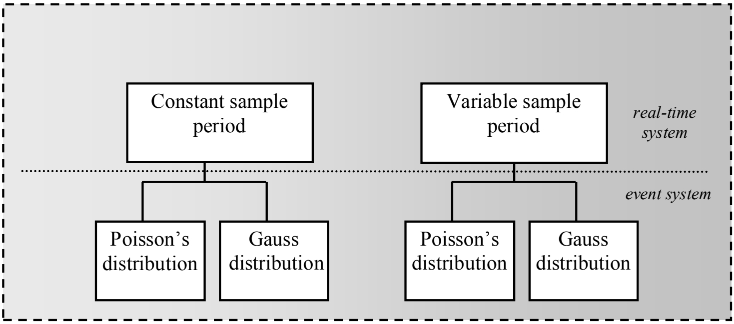


Fig. 5 *Methods of analysis and synthesis of combined dynamic systems*

1.1 Interruptions

When implementing the control program, a program handling some events occurring in the system stochastically must be ensured. Most of these events might be tested within the transferred program; yet it is more effective if the event is handled by the processor at the level of technology without polluting the program in each step of testing all the permissible options for the generation of events. The servicing of new events is performed by processor interruptions. Interruption is an event triggered by an external stimulus or a corresponding program instruction, whereby the processor interrupts the currently executed program, performs a service program designated by the interrupt vectors, and, after its execution, goes back to that part of the original program where it was interrupted.

Most of the current automation control tools (e.g. programmable logic controllers) work in a cycle period with the following sequence [23]:

1. the operating system initiates a cyclic time processing,
2. the processor copies the values of processing the output image into the output module,

3. the processor reads the status of the input modules and then restores and adjusts input image processing,
4. the processor processes the user (control) program in the determined time and performs the programming instructions,
5. in the end of cycle, the operating system executes the tasks listed (uploading and deleting blocks),
6. the processor returns to the beginning of the cycle and resumes the cyclic processing.

The sequence of items 2-5 is cyclically repeated in runtime applications (Fig. 6) [23].

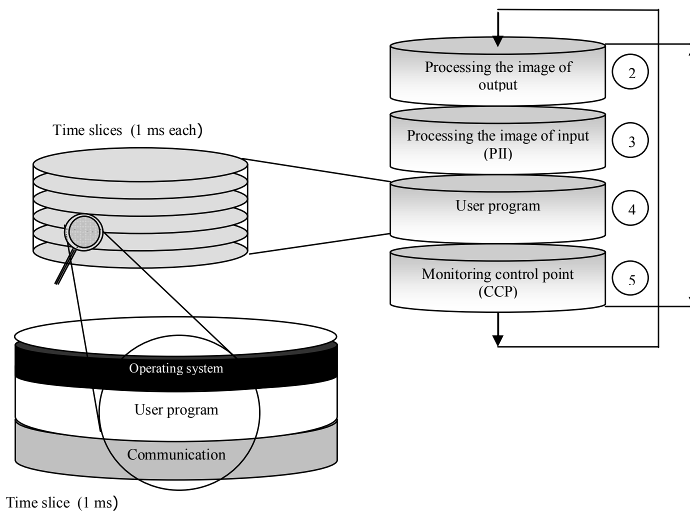


Fig. 6 *Example of processing cycle in PLC*

The following factors influence the duration of the processing period [23]:

- time necessary to execute inevitable functions of operating system,
- transfer of processing the images (images of inputs and outputs),
- period of executing the user program,
- timer of processor,
- time necessary for communication (e.g. via PROFIBUS, MPI etc.),
- integrated functions,
- occurrence of interruptions – hardware or software events in the system.

In automation systems, interruptions are implemented via the organisation blocks. An organisation block (OB) represents the interface between the operation system and the user program. The operating system activates the performance of the given organization block controlling the cyclic program execution including possible interruptions (if there are requirements for the interruptions). Organisation blocks are interrupted based on the organisation blocks' priority, while the OBs with higher priority may interrupt the execution of the OB with lower priority. Appendix A contains the description of several basic types of interruptions and corresponding organisation blocks according to the Siemens S7-300 and S7-400 tools [24]. The period of cyclic execution of Tcyc is not equal for each cycle; its duration changes in dependence on the number of stochastic interruptions (organisation blocks) in each cycle, as illustrated in Fig. 7, where period Tcyc1 is smaller than period Tcyc2, regarding the number of OB10 interruptions' occurrence during the cycle program execution.

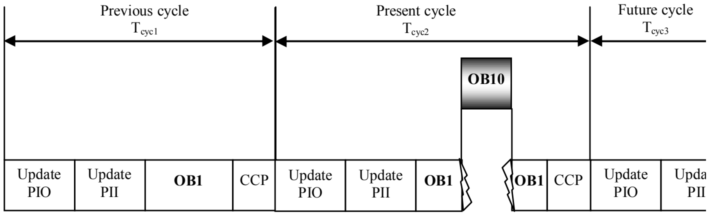


***Fig.** 7 Prolongation of the cycle execution via interruptions in OB10*

Average prolongation of a processing cycle in dependence of the type of interruption is illustrated in Table 1. Its size depends on the particular type of processor [23].

MEAN DISTRIBUTION OF THE CYCLE AT THE OCCURRENCE OF RELATED TYPE OF INTERRUPTION

Table 1

Type of interruption	**Type of CPU**					
	313	314	315	315-2 DP	316-2 DP	318-2
Process interruption	700 µs	700 µs	480 µs	590 µs	590 µs	do 340ms
Diagnostic interruption	880 µs	880 µs	700 µs	860 µs	860 µs	450 µs
Interruptions in day-time	-	680 µs	460 µs	560 µs	560 µs	350 µs
Interruptions due to delay	-	550 µs	370 µs	450 µs	450 µs	260 µs
Interruptions in intervals	-	360 µs	280 µs	220 µs	220 µs	260 µs
Program, access errors / errors of program execution	740 µs	740 µs	560 µs	490 µs	490 µs	285 µs

The minimum and maximum responses of the system to the occurring events are illustrated in Table 2 [23].

MINIMUM AND MAXIMUM RESPONSE OF PROCESSOR TO INTERRUPTION

Table 2

Typ CPU	Min. [ms]	Max. [ms]
313	0.5	1.1
314	0.5	1.1
315	0.3	1.1
315-2DP	0.4	1.1
316-2DP	0.4	1.1
318-2	0.23	0.27

2. DETERMINING A SAMPLE PERIOD IN CDS

A control dynamic system works with the sample period T_O (cyclic processing) corresponding to the Shannon-Kotelnikov theorem and the time necessary to process all the necessary processes in each execution cycle.

In dynamic systems, the considered sample period T_O is generally constant, comprising just the time T_R necessary to process all the necessary control and cyclically repeated processes in a particular system. In combined dynamic systems, however, the sample period in question is enhanced by the time constant T_P necessary to execute random events in the system, regardless of whether it is constant or variable (Fig. 8).

$$T_O = T_R + T_P \tag{1}$$

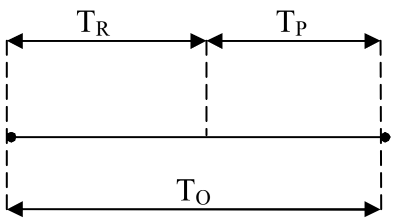


Fig. 8 *Sample period T_O*

In order to determine time T_P necessary to perform service routines of random events in a particular cycle, I used statistical evaluation of arising events and their probability estimate, as well as the determination of the most probable time constant T_P, by using some of the probability distribution laws. The following chapters will discuss in detail the determination of the event time constant of random events T_P of sample period T_O by means of two basic laws of probability distribution (Gauss and Poisson's laws), while the selection of the proper one depends on the particular application and approximation of arising events to one of the aforementioned laws.

2.1 Normal probability distribution when searching for the event time constant

In normal probability distribution, the following uniform distribution of arising events around the average value Ta calculated from the following equation is supposed:

$$T_a = \frac{T_c}{N}, \tag{2}$$

where T_c is the time necessary to service all the arisen interruptions, and N is the number of the interruptions occurring in a given cycle.

The average quadratic deviation from the mean value T_a can be then determined by the relation:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n-1}}, \tag{3}$$

where $\mu = T_a$ – the arithmetic average of the time necessary to service all the arisen interruptions.

The criteria for processing randomly arising interruptions, i.e. the probability of processing the events arising in a particular sample period in a combined dynamic system, can be determined by using the average quadratic deviation. The size of the sample period in such case corresponds to the mean value of time μ; the probability that the events arisen in this interval prolonged/reduced by the time representing mean quadratic deviation, then equals (Fig. 9):

- 68.3% - at μ+/-σ,
- 95.6% - at μ+/-2σ,
- 99.7% - at μ+/-3σ.

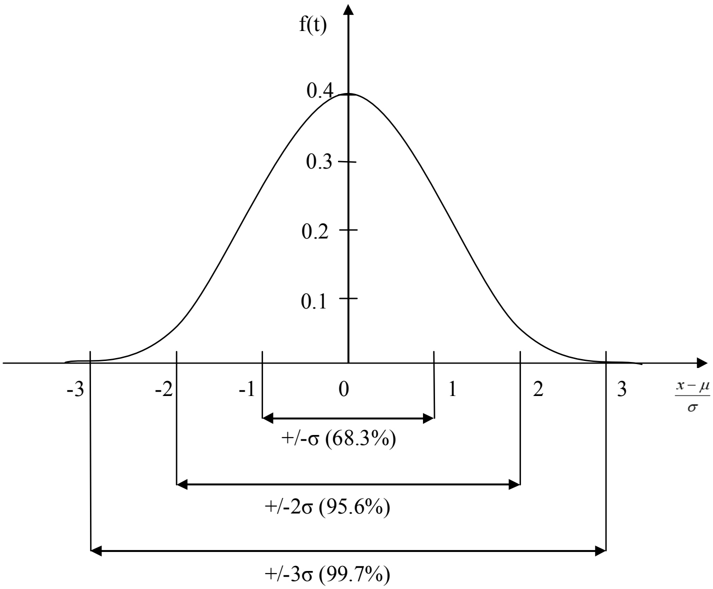


Fig. 9 *Density of probability distribution at standardised normal distribution*

The density of probability distribution f(t) is a derivation of distribution function F(t) enabling to determination of the probability of generation of a certain number of events nx, as well as the probability of occurrence of that number of events in time interval Δt [25]:

$$f(t) = \frac{dF(t)}{dt} \approx \frac{n_x(t+\Delta t) - n_x(t)}{N\Delta t} \tag{4}$$

The density of probability distribution f(t) in time t takes the following form [26]:

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma}}, \tag{5}$$

while its maximum corresponds to the value of function f(t) in time Ta (Fig. 10)

$$f(T_a) = \frac{1}{\sigma\sqrt{2\pi}}. \tag{6}$$

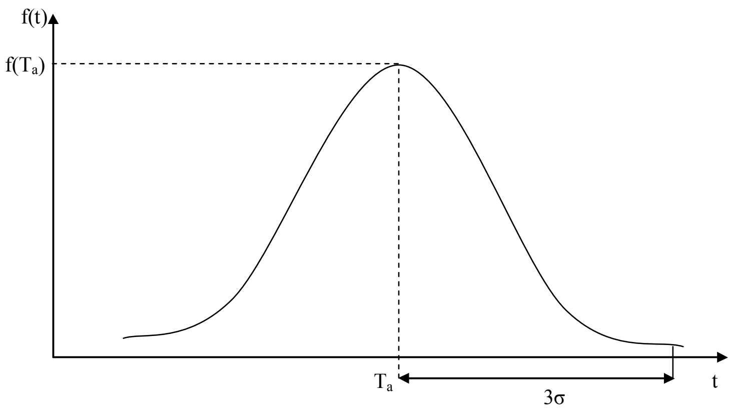


Fig. 10 *Density of probability distribution*

Time constant T_p in a combined system can be equal to:

a) the average time value necessary to service the arising events => $T_p = T_a$,

b) the average time value necessary to service the arising events increased/decreased by value Δ => $T_p = T_a$ +- Δ, including the time necessary to service the events with the defined percentage of probability (e.g. +σ representing probability approximately 74%),

c) the average time value necessary to service arising events and a triple of mean quadratic deviation $T_p = T_a + 3_\sigma$, while this value includes the service of all events arisen in the cycle with probability of 99.7%.

Supposing that the time T_A necessary to execute all events in a combined dynamic system is:

1. $T_A < T_O$ – smaller than total sample period T_O, while
   a) $T_R >> T_A$ – constant of time-activated system T_R is much bigger than the time necessary to execute all events T_A; time T_A is then negligible,
   b) $T_R \approx T_A$ – constant of time-controlled system T_R is approximately equal to the time necessary to service all events T_A,
   c) $T_R << T_A$ – time T_A necessary to service all events is much longer than constant T_R of a time-controlled system.
2. $T_A > T_O$ – is bigger than the total sample period; minimum number of the steps K_{min}, necessary to perform the service can be then determined from the relation

$$K_{min} \geq T_A/T_{p,} \tag{7}$$

or $T_p=T_A/K_{min}$, while $T_A=T_{a1}+T_{a2}+.....+T_{ak}$.

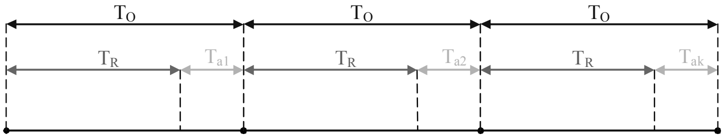


***Fig. 11** Disintegration of T_A into several steps regarding T_P*

In the last method, the total time necessary to execute all service routines is distributed to more cycles according to the value of step K_{min} (Fig. 11).

2.2 Power spectrum of stochastic signal

In harmonic analysis, deterministic processes can be expressed by means of a Fourier series or Fourier integral, depending on whether periodic or aperiodic processes are concerned. Energy distribution in stationary random processes (their internal structure, resp.) can be similarly characterised by using correlation analysis. The procedure can be then applied to optimise a spectral analysis of dynamic properties of combined dynamic systems.

A correlation function determines the rate of dependence or similarity of signals. If analysing just one signal in various time moments, an auto-correlation function is implied. For periodic signals with period T, the auto-correlation function takes the following form:

$$K(\tau) = \frac{1}{T}\int_{-T/2}^{T/2} f(t)f(t-\tau)dt \quad . \qquad (8)$$

For non-periodic signals, it can be expressed in the following form:

$$K(\tau) = \int_{-\infty}^{\infty} f(t) f(t-\tau) dt \quad . \tag{9}$$

Based on Wiener-Chinchin relations, it is possible to express the mutual relation between the auto-correlation function Kx(τ) of the stationary stochastic process and its power spectral density Sx(ω) [27]. Supposing that the following conditions of integration are met,

$$\int_{-\infty}^{\infty} K_x(\tau) d\tau \leq M \quad , \quad \int_{-\infty}^{\infty} S_x(\omega) d\omega \leq N \quad , \tag{10}$$

where M and N are arbitrary finite values, then the above-mentioned functions form a couple in Fourier transform:

$$K_x(\tau) = \frac{1}{4\pi} \int_{-\infty}^{\infty} S_x(\omega) e^{j\omega t} d\omega \quad , \tag{11}$$

$$S_x(\omega) = 2 \int_{-\infty}^{\infty} K_x(\tau) e^{-j\omega\tau} d\tau \quad . \tag{12}$$

Since the auto-correlation function is real and even, it can be written as

$$K_x(\tau) = \frac{1}{2\pi} \int_{0}^{\infty} S_x(\omega) \cos(\omega t) d\omega \quad . \tag{13}$$

Similarly, the function of spectral density can be expressed:

$$S_x(\omega) = 4 \int_{0}^{\infty} K_x(\tau) \cos(\omega t) d\tau \quad . \tag{14}$$

If τ=0, the resulting formula

$$K_x(0) = D_x = \frac{1}{2\pi} \int_{0}^{\infty} S_x(\omega) d\omega \quad , \tag{15}$$

then expresses the total mean standardised power of stationary stochastic signal. Similarly, spectral power density for ω=0 can be expressed:

$$S_x(0) = 2\int_0^{\infty} K_x(\tau)d\tau \quad , \tag{16}$$

indicating that the surface under the curve of the autocorrelation function is proportional to the value of the power spectral density for ω=0. The properties of the Fourier transform also suggest that the narrower the frequency of the power spectral density, the wider the time region of the corresponding correlation function. Hence, if narrowing the spectrum by filtration, the correlation between distant values of the random process increases; and, vice-versa, if the power spectrum is indefinitely wide (e.g. white noise), the auto-correlation function will be indefinitely narrow, and the closest values in such a random process will be uncorrelated.

2.3 White noise

The theory of white noise can be applied to combined dynamic systems, while the stochastic part of the systems, approximating the normal probability distribution, is represented as a process of constant and frequency-unlimited intensity of energy spectral density. White noise colouring is supposed inside the system itself. Using the assumptions of normal probability distribution, we can derive the value of the Gaussian white noise intensity, a stochastic event element of combined dynamic systems, respectively.

White noise is a stationary random process, the power spectral density of which is constant within the whole frequency range (Fig. 12) [28]:

$$S_x(\omega) = N = const. \tag{17}$$

The autocorrelation function of white noise, the mean power of which is infinite, can be then written according to Wiener-Chinchin relations as:

$$K_x(\tau) = \frac{N_0}{2\pi} \int_{-\infty}^{\infty} e^{j\omega\tau} d\omega = N\delta(\tau) \tag{18}$$

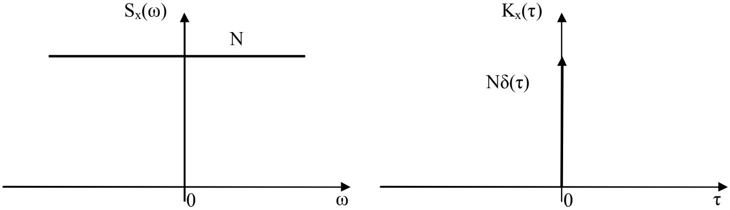


Fig. 12 *Power spectral density and auto-correlation function of white noise*

Dispersion of the frequency-limited white noise can be written in the following form [29]:

$$D = \sigma^2 = \frac{1}{2\pi} \int_{-\omega_m}^{\infty_m} N d\omega = \frac{N\Delta\omega}{2\pi} = N\Delta f$$, where $\Delta\omega = 2\omega_m$. (19)

The mean quadratic error σ then equals

$$\sigma = \sqrt{N}\sqrt{\Delta f}, \tag{20}$$

while for frequency-limited white noise, auto-correlation takes the function

$$K_x(\tau) = \frac{1}{\pi} \int_0^{\omega_m} N \cos\omega\tau d\omega = \frac{N}{\pi\tau} \sin\omega_m \tau. \tag{21}$$

In a normal probability distribution, the dispersion of the stochastic process equals $D = 6\sigma$; after substituting into equation (19), we get the expression of random process intensity:

$$N = \frac{D}{\Delta f} = \frac{6\sigma}{\Delta f}\,. \tag{22}$$

White noise is a theoretical assumption; it is colourful noise practically providing input into the system. In this particular case, system input in the form of stochastic events is decoloured, thus supposing that event generation is subject to the white noise requirements, while colouring is transferred to the system itself (Fig. 13).

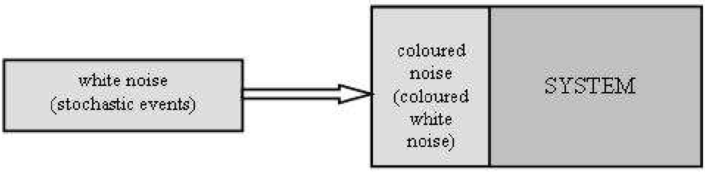


Fig. 13 *White noise colouring inside the F system*

2.4 Poisson's probability distribution

In case the events generation approaches Poisson's probability distribution, the theory can be used to determine the probability of interruption occurrences in a given time interval, or such a time constant of the event element of combined systems, in which the events would be processed according to the determined probability. Poisson's distribution may comprise dynamics of arising events (energy spectrum of combined systems).

Poisson’s distribution is based on the following binomic probability distribution [27]

$$P_n(k) = \binom{n}{k} p^k q^{n-k} \quad , \tag{23}$$

where k-number of events at n-independent attempts, p – examined value, q=1-p.

By means of the Stirling formula

$$m! \approx \sqrt{2\pi m} . m^m . e^{-m} \tag{24}$$

expressing the approximate value of a factorial with the precision increasing with the increasing number from which the factorial is calculated (Tab. 3) [30], and further modifications of binomic distribution, we can get the expression asymptotically approaching a binomic distribution, the so called Laplace formula

$$P_n(k) \approx \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\delta_k^4}{2\sigma^2}} \text{, where } \delta_k = k - [(n+1)p] \text{, } \sigma = \sqrt{npq} \text{.} \tag{25}$$

PRECISION OF CALCULATING FACTORIAL BY MEANS OF STIRLING FORMULA

Table 3

No.	Factorial	Stirling formula	Error in %
1	1	0.922	8
2	2	1.919	4
5	120	118.019	2
10	3.6288 · 106	3.5986 · 106	0.8
100	9.3326 · 10157	9.3249 · 10157	0.08

The increasing number of phenomena studied by using the Laplace formula leads to inaccurate results, however. The second asymptotic probability distribution, the so-called Poisson's probability distribution, is therefore used. Binomic calculation of probability can be expressed in the following format [27]:

$$P_n(0) = (1-p)^n = (1-\frac{\lambda}{n})^n \quad , \tag{26}$$

where $p = \dfrac{\lambda}{n}$ is the probability of λ-phenomena occurrence out of the total number of possible events n.

After modification, we get the form

$$ln\, P_n(0) = n.ln(1-\frac{\lambda}{n}) = -\lambda - \frac{\lambda^2}{2n} .... \tag{27}$$

If

$$\frac{\lambda^2}{n} << 1, \text{ or } \quad p << \frac{1}{\sqrt{n}}, \tag{28}$$

we can focus only on the first sequence member in the previous expression, getting

$$P_n(0) \approx e^{-\lambda}. \tag{29}$$

Analogically, we can write

$$P_n(1) = np(1-p)^n = \frac{np}{1-p}(1-p)^n = \frac{\lambda}{1-\frac{\lambda}{n}} P_n(0) \approx \lambda e^{-\lambda} \quad , \tag{30}$$

and in general form

$$P_n(k) = \frac{n(n-1)...(n-k+1)}{k!} p^k (1-p)^{n-k} = \frac{n(n-1)...(n-k+1)}{n^k(1-\frac{\lambda}{n})^k} \frac{\lambda^k}{k!} P_n(0) \approx \frac{\lambda^k}{k!} e^{-\lambda} \tag{31}$$

The probability that just the k-phenomena with the mean parameter λ is given by Poisson's relation

$$P_n(k) \approx p(k,\lambda) = \frac{\lambda^k}{k!}e^{-\lambda} \quad , \tag{32}$$

and cumulative Poisson's function supposing the occurrence of not more than m-elements with a mean parameter λ, acquires the form [27]:

$$P(k \le m) = \sum_{k=0}^{m} P_n(k) \approx P(m,\lambda) = \sum_{k=0}^{m} \frac{\lambda^k}{k!}e^{-\lambda} \quad , \tag{33}$$

or

$$P(m,\lambda) = e^{-\lambda}\sum_{k=0}^{m}\frac{\lambda^k}{k!} = \frac{1}{m!}\int_{\lambda}^{\infty} z^m e^{-z} dz \quad . \tag{34}$$

Poisson's asymptotic distribution supposes an equal probability of random and mutually independent events. The general formula for determining the probability of x-interruptions is given by the relation

$$P(x) = e^{-\lambda}\frac{\lambda^x}{x!} \quad , \tag{35}$$

where λ is the mean Poisson's parameter.

After introducing the substitution $\lambda = \nu t$, where ν represents the mean number of interruptions in a time unit, and t represents the time examined (e.g. time necessary to implement one interruption or n-interruptions), and after substitution into the original formula, we can calculate the probability of just x-events generation in the time interval t:

$$P(\upsilon t) = e^{-(\upsilon t)}\frac{(\upsilon t)^x}{x!} \quad . \tag{36}$$

The aforementioned equation allows for expressing:

- the calculation of the time event constant T_P for defined probability of interruptions occurrence and maximum number of interruptions x at the average number of interruptions v,
- the calculation of the maximum number of interruptions x which can be processed in the interval corresponding to the defined event time constant T_P and defined probability P.

In Poisson's distribution, having determined the mean value of the sample period of combined dynamic systems, the following hypotheses can be postulated:

1. $T_A < T_O$ – the time necessary to process all events in the system is shorter than the total sample period, while:
   a) if the time necessary to service all events is too short when compared to the constant of time-driven system $T_A << T_R$, the event element of sample period may be then neglected; hence, $T_O = T_R$,
   b) if $T_R \approx T_A$, i.e. the constant of a time-driven system is approximately equal to the time necessary to service all events in system,
   c) $T_A >> T_R$ – the time necessary to service all events is much longer than the constant of the time-driven system.
2. $T_A > T_O$ – the time necessary to process all events is longer than sample period, the minimum number of steps K_{min}, in which all the generated events will be implemented, may be then derived from the relation (7).

2.5 Energy spectrum of combined systems

To specify the energy expression of stochastic events in combined systems generated according to the Poisson's distribution, we can use an analogy from the field of electro-technology related to the transfer of a telegraph signal [27].

A signal (Fig. 14) represents the arising events with amplitude S(t) in time t. If an interruption signalling the events in the generated the system is detected, the signal amplitude changes to the value h and then stays constant within the whole period of the running service programs related to stochastic events.

Each new event in a system is included in the buffer of events listing all the events waiting for the performance of appropriate service programs. The order of the events in the buffer depends on the time of their generation and priority assigned to individual interruptions. In case no new interruption was registered in the system, or there is no new event in the buffer, signal amplitude will stabilise at value 0. Execution time of service programs of events in a particular moment corresponds to time tu. In case no new event was registered in the system, periodic execution of the time-activated part of combined systems will take place.

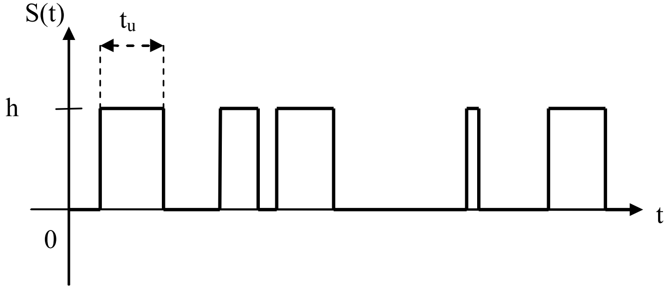


Fig. 14 *Event signal in combined systems*

This theorem assumes stochastic events generation corresponding to Poisson's distribution (Fig. 15).

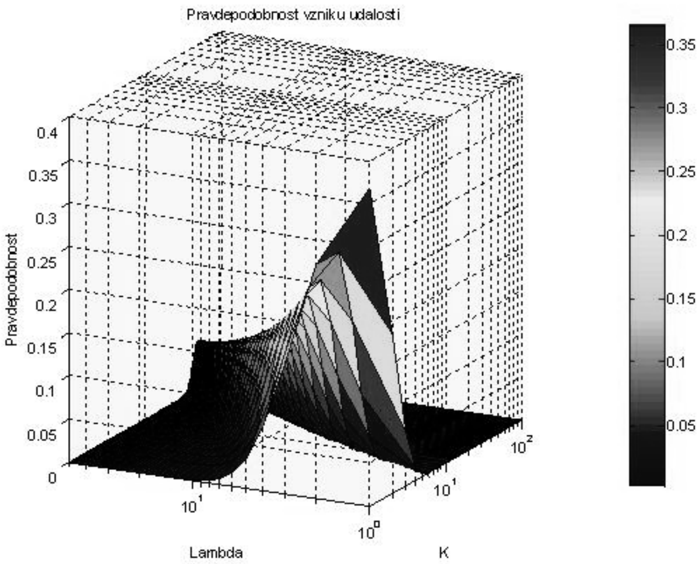


Fig. 15 *Poisson's probability distribution*

If the mean value of the occurrence of events is denoted as v, then the product vt represents the mean value of the events occurrence at time t. To calculate the Poisson mean parameter, i.e. the mean value of the events occurrence over any period of time $<t_1, t_2>$, we use the following formula

$$\lambda = \int_{t_1}^{t_2} v(t)dt \quad . \tag{37}$$

The autocorrelation function of the signal presented between two time moments t_1, t_2 corresponds to the expression

$$K(t_1, t_2) = h^2 e^{-2\int_{t_1}^{t_2} v(t)dt} \quad . \tag{38}$$

In the case of a stationary signal, i.e. $v(t)=v_0=const$, the correlation function depends on the difference of these two time moments, $\tau=t_2-t_1$, then

$$K(\tau) = h^2 e^{-2v_0|\tau|} \quad . \tag{39}$$

Wiener-Chinchin relations can be used to express the corresponding energy spectrum of the stationary random signal on the basis of the autocorrelation function

$$F(\omega) = 4h^2 \int_0^{\infty} e^{-2v_0\tau} \cos\omega\tau d\tau = \frac{8h^2 v_0}{\omega^2 + 4v_0^2} \quad . \tag{40}$$

The autocorrelation function and its spectral density of energy distribution are illustrated in Fig. 16.

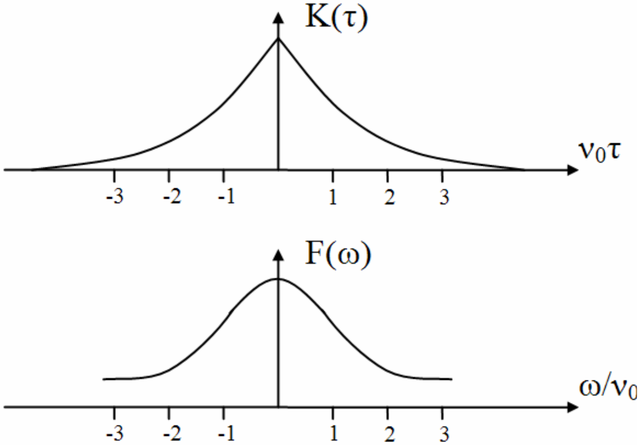


Fig. 16 *Autocorrelation function K(τ) and corresponding spectral density F(ω)*

Instead of signal amplitude, let us assume the probability that an event has or has not occurred in the system. In the case that an event is registered in the system and related processes are executed, h gets the value 1. After substituting h=1, the autocorrelation function equals

$$K(\tau) = e^{-2\nu_0|\tau|}. \tag{41}$$

Its expression for various values ν0 varies regarding the dependence shown in Fig. 17.

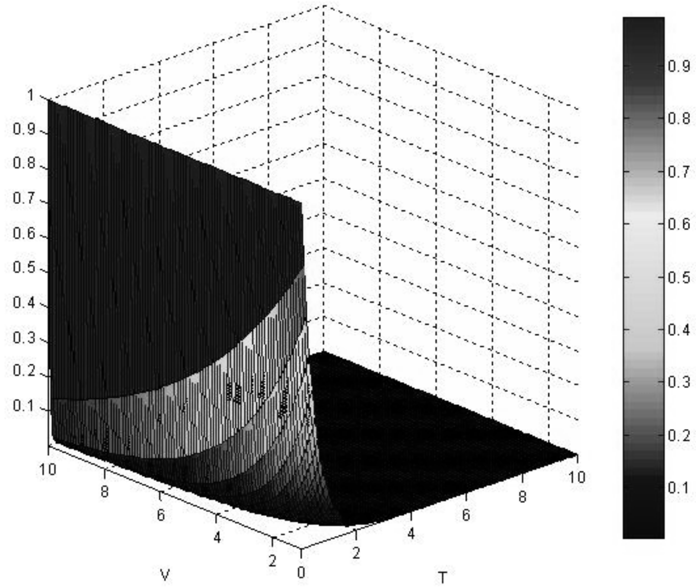


Fig. 17 *Autocorrelation function K(τ) for various values of* v_0

After substituting probability h into the original formula of energy distribution density, we obtain the density of the energy distribution probability of random signal in the form

$$F(\omega) = \frac{8v_0}{\omega^2 + 4v_0^2}\,. \tag{42}$$

The density of probability distribution at the mean value λ and the number of events k is graphically displayed in Fig. 18.

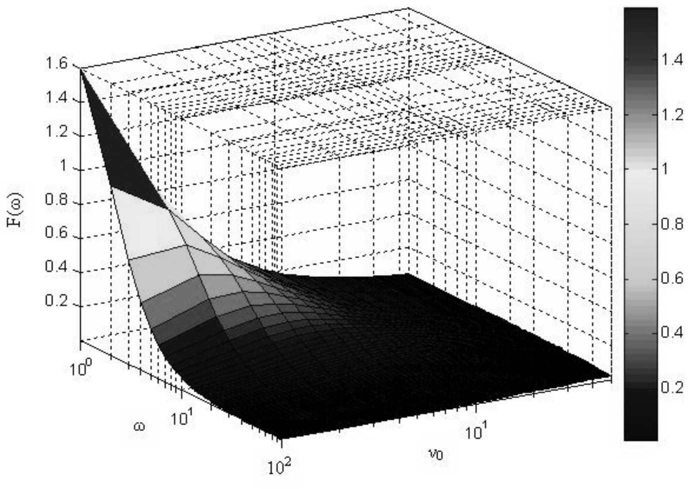


Fig. 18 *Density of probability distribution F(ω)*

In the case of a non-stationary process, the correlation function will depend on two variables; its form then equals

$$K(\tau) = h^2 \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} e^{-2\lambda(t,\tau)} dt \quad . \tag{43}$$

An average number of changes of events in time then equals

$$v(t) = v_0 + v_1 \cos \omega_0 t \quad . \tag{44}$$

The increase of changes of events in time τ is then equal to function λ(t,τ)

$$\lambda(t,\tau) = \int_{t}^{t+\tau} (v_0 + v_1 \cos\omega_0 t)dt = v_0\tau + \frac{v_1}{\omega_0} 2\sin\frac{\omega_0\tau}{2}\cos(\omega_0 t + \varphi) \quad . \qquad (45)$$

This function is periodic, with the period T=2π/ω0. The autocorrelation function of non-stationary signal then equals

$$K(\tau) = \frac{h^2 e^{-2v_0\tau}}{T}\int_{0}^{T} e^{-\frac{4v_1}{\omega_0}\sin\frac{\omega_0\tau}{2}\cos(\omega_0 t+\varphi)}dt = \mathrm{h}^2\mathrm{e}^{-2v_0|\tau|} I_0(\frac{4v_1}{\omega_0}\sin\frac{\omega_0\tau}{2}) \quad . \qquad (46)$$

The Bessel function can be expressed in the Fourier series

$$I_0(\frac{4v_1}{\omega_0}\sin\frac{\omega_0\tau}{2}) = \sum_{k=-\infty}^{\infty} c_k e^{ik\omega_0} \quad , \qquad (47)$$

where coefficient ck is equal to

$$c_k = c_{-k} = \sum_{n=k}^{\infty} (-1)^k (\frac{v_1}{\omega_0})^{2n} \frac{1}{(n!)^2} \frac{(2n)!}{(n-k)!(n+k)!}, \text{ where k>0.} \qquad (48)$$

The result of the Fourier autocorrelation function is the density of energy distribution in the form of Fourier series

$$F(\omega) = \sum_{k=-\infty}^{\infty} F_k(\omega) \quad , \qquad (49)$$

where

$$F_k(\omega) = \frac{8h^2 c_k v_0}{(\omega - k\omega_0)^2 + 4v_0^2} \quad . \qquad (50)$$

Like in a stationary random process, we consider that instead of the amplitude of the transient signal into quantity h, we substitute probability corresponding to events generation in system. If the system registers an event, then h=1, and the density of energy probability will be equal to

$$F_k(\omega) = \frac{8c_k v_0}{(\omega - k\omega_0)^2 + 4v_0^2} . \tag{51}$$

2.6 Conclusion

From the aspect of analysis, synthesis or optimisation of control systems generated in co-operation of discrete systems of real time and event discrete systems representing a combined discrete dynamic system, it is recommended to determine the sample period so that (besides cyclically recurring control processes) the control system is able to register and process the stochastic events arising in the system.

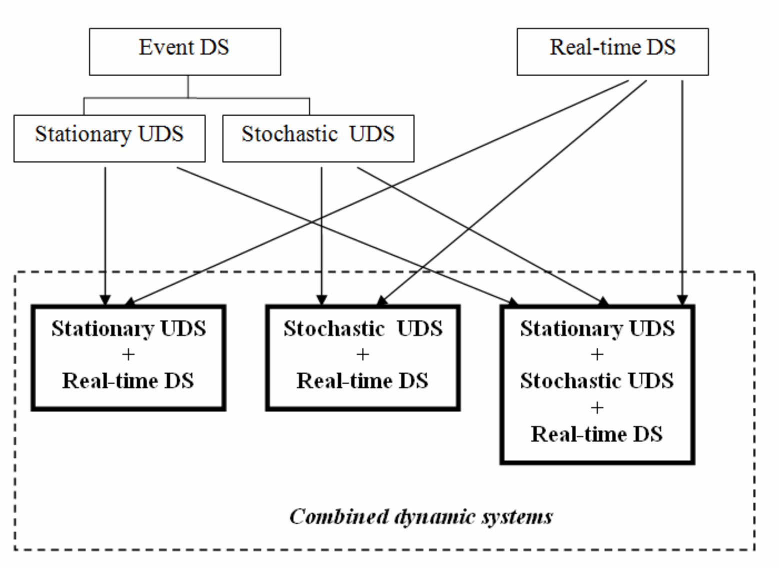


Fig. 19 *Combined dynamic systems*

Cooperation of both systems can be split into three basic combined dynamic systems (Fig.19):

1. stationary event dynamic system + dynamic real-time system,
2. stochastic event dynamic system + dynamic real-time system,
3. stationary + stochastic event system + dynamic real-time system.

Given the existence of two mathematically and functionally different systems, the sample period of combined dynamic systems TO is determined by the sum of the sub-periods of the two systems, namely:

$$T_O = T_R + T_P, \tag{52}$$

where T_R – sample period of dynamic time-controlled system,

T_P – period of cyclic processing of the event dynamic system.

When classifying combined dynamic systems, it is necessary to distinguish two types, depending on the period of cyclic processing of the event element of combined systems:

- with constant sample period, where $\mathrm{T_P} = const \Rightarrow \mathrm{T_O} = const$,
- with variable sample period, where $\mathrm{T_P} = var \Rightarrow \mathrm{T_O} = var$.

Determination of the sample period of real-time system TR is based on the verified and generally recognised procedures of analysis and synthesis of the systems based on mathematical description of a dynamic system. The sample period is determined by the well-known laws and theorems.

When determining the period of cyclic processing of stationary event dynamic systems, it is possible to start from the basic transient systems and some of the models generated by verified and standardised model tools. Deterministic Petri nets, where the period determination is the search of the longest ways and summing the partial periods corresponding to individual places and transitions, seem to be the most suitable of the tools.

One of the options for determining the time necessary to process stochastic events T_P (with defined criteria and requirements) is the application of statistics and probability, either by means of normal (Gauss) or Poisson's probability distribution. Determination of the event time constant depends either on the selected criteria (its size depends on the selection of the probability value of the arising events processing), or on the way of the implemented control (e.g. if all events must be performed in the given period, or may be transferred to the following sample period).

For the purposes of analysis and synthesis, it is important to know the dynamic properties of such systems. Application of a normal probability distribution to the event part of combined systems can comprise the statistics describing the system behaviour (by using white noise theory and the power spectrum of a stochastic system), while the application of Poisson's distribution of energy probability spectrum also enables description of the dynamics of events generation probability in a system.

3. ALGORITHMS OF CONTROL AND ANALYSIS OF STOCHASTIC ELEMENT OF SAMPLE PERIOD

This chapter:

a) discusses the applied procedure of evaluating the spectral analysis of a model system via simulation in the environment,

b) describes in detail the designed adaptive algorithm of events processing, as well as the determination of a sample period of combined discrete dynamic systems,

c) introduces the verification models of combined dynamic systems for both Gauss and Poisson's probability distribution, which were used to analyse the impact of stochastic events on the whole control system described by differential equations. The verification process was carried out via a program generated in C language and the results in the form of transient characteristics were presented in the form of graphs.

3.1 Determining the impact of stochastic sample element via spectral analysis

The impact of varying the total sample period and the time necessary to process and execute all control interventions and cyclically recurring instructions bound to the real part of combined systems in dependence on stochastically changing time necessary to service the events generated in a particular cycle were simulated by means of Matlab or Simulink. A simple continuous system of the second grade was assumed with feedback and transient function in the form:

$$W(p) = \frac{1}{Tp^2 + p + 1} \tag{53}$$

where $T=1s$.

After Z-transformation of continuous dynamic system to discrete one, we get the impulse transient function

$$W(z) = \frac{T_0 z^{-1}}{1 - z^{-1}} - 1 + \frac{1 - z^{-1}}{1 - z^{-1} e^{-T_0}}, \tag{54}$$

while the discrete transient function for $T_0=0.1s$ acquires the form

$$W(z) = \frac{0.1 z^{-1}}{1 - z^{-1}} - 1 + \frac{1 - z^{-1}}{1 - 0.9048 z^{-1}}. \tag{55}$$

The system model with the aforementioned transient function is displayed in Fig. 20.

The simulation model generated in Simulink with the sample period $T_0=0.1s$ excited by a harmonic signal with circular frequency $\omega = 1\ rad/s$, amplitude $A=1$ and a variable fraction delay z^{-n} (representing the random shift of input signal samples influenced by stochastically generated events) is shown in Fig.21.

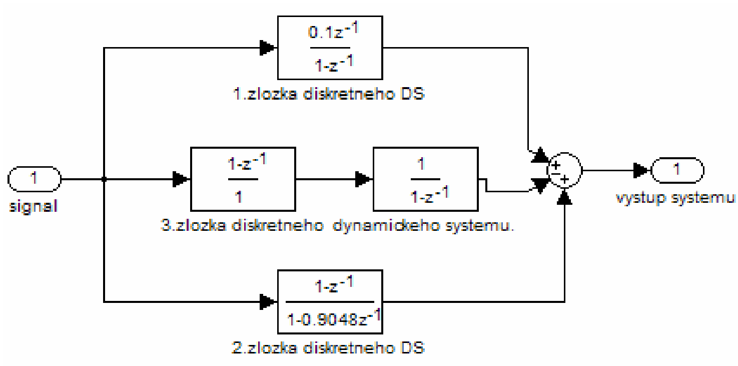


Fig. 20 Model of discrete dynamic system

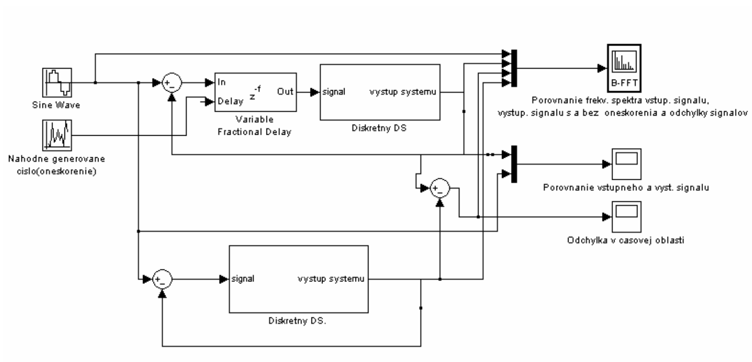


Fig. 21 Simulation model with variable fraction delay

Besides the system with a randomly generated fraction delay of input samples of the harmonic signal, the model comprises an additional identical discreet dynamic system with the aforementioned impulse transient function, yet without stochastic delay. Comparing the outputs of these two systems in time and frequency regions, we obtain the system deviation caused by randomly generated fraction delay representing stochastically arising events in a combined dynamic system. The simulation was carried out for four maximum values of randomly generated numbers of delays: *+/-2*T_0, *+/-4*T_0, *+/-8*T_0, *+/-16*T_0. In the process of system analysis, we used the expression of frequency spectra for:

- the input harmonic signal,
- the output signal from a discrete system with random fraction delay,
- the output signal from discrete system without delay,
- the output deviation of both signals (randomly generated delay).

The designed scheme of modelling the impact of the random component of the sample period by means of spectral analysis enables detection of its influence on the control system behaviour. As seen in Fig. 22, an increased random component of the sample period increases the spectrum of the error element, which may provide the basis for assessing the influence of the stochastic element on control quality.

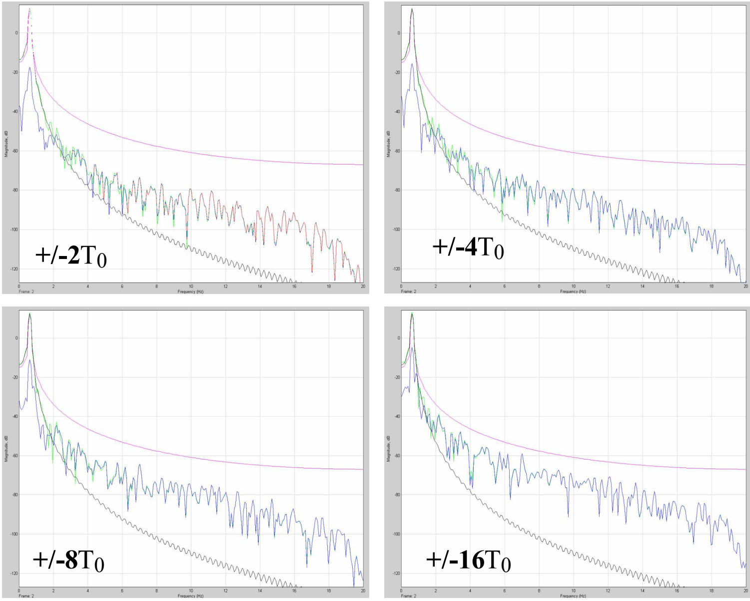


Fig. 22 *Frequency spectrum of input harmonic signal (violet colour), output signal with variable delay (green colour), output signal without delay (black colour) and difference of output signals with/ without stochastic delay (blue colour) for four different maximum values of randomly generated numbers (+/-2T_0,4T_0,8T_0,16T_0)*

3.2 Design of adaptive algorithm of processing events and determination of sample period of combined dynamic systems

Figure 23 displays the designed methodology of the control algorithm for combined dynamic systems in the form of an adaptive algorithm of the events processing implementation and auto-correction of the stochastic time element of the combined system with a variable sample period. The submitted algorithm consists of two parts: declaration and program.

Besides variables initialisation, the declaration part comprises:

- calculation of the event component T_{Podhad} of the regulator sample period based on the related law of probability distribution,
- setup of the sample period of the control member, i.e. a programmable controller. The sample period is given by the size of T_R, i.e. the time necessary to execute cyclical program instructions of the time-controlled part of a combined dynamic system and the time T_{Podhad} necessary for the implementation of the event part of a combined dynamic system.

Legend to Figure 23:

N – *const., specifies max. number of consecutive sample periods with the value exceeding double of probability-determined event time element* $2T_{Podhad}$,

Number – *auxiliary variable incrementing real number of periods with the value bigger than 2* T_{Podhad},

Fr_1 – *event queue No. 1,*

Fr_2 – *event queue No. 2,*

T_{Podhad} – *probability-determined event time element of sample period of regulator* T_{REG},

T_R – *constantly determined time element of sample period for execution of cyclic instructions of time-controlled part of combined dynamic system,*

T_{REG} – *sample period of programmable regulator,*

T_P – *real time corresponding to execution of program blocks and instructions of individual implemented events,*

τ_i – *time of executing service routines of the* i^{th} *event*,

K – *auxiliary variable distinguishing the change of active queue already implemented,*

Change –*auxiliary variable signalling the change of active queue.*

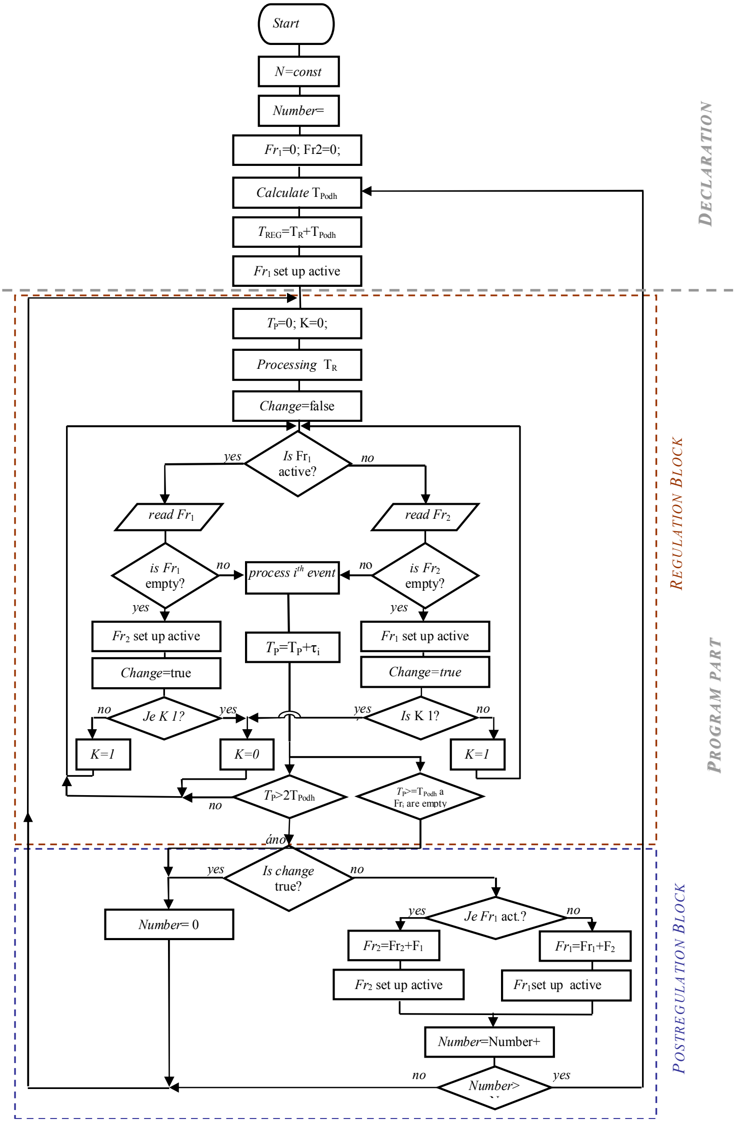


Fig. 23 Design of algorithm for processing the events of combined systems

The mean value of a variable sample period is given by the addition

$$T_0 = T_R + T_{Podhad}\,. \tag{56}$$

Since the estimate of the event time constituent of the sample period is centred, and the aforementioned variant of the algorithm is designed to perform all the statistically assumed events generated in the system, the maximum value of sample period is defined by the form

$$T_{0max} = T_R + 2T_{Podhad}\,. \tag{57}$$

The particular value of the aforementioned sample period in a given control process is fully dependable on the number of generated events in the system and the related service routines (Fig. 24).

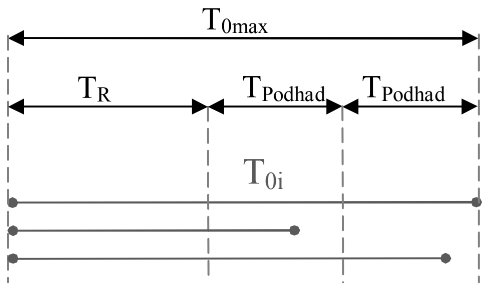


Fig. 24 *Sample period in control process*

The program part first processes instructions of the time-controlled part of the system. All events generated in the system are arranged in the relevant queue of events according to the related algorithm of planning and scheduling tasks. Two queues are used for the control implementation. The events arranged in an active queue are gradually serviced, and the related service program blocks are simultaneously executed (Fig. 25).

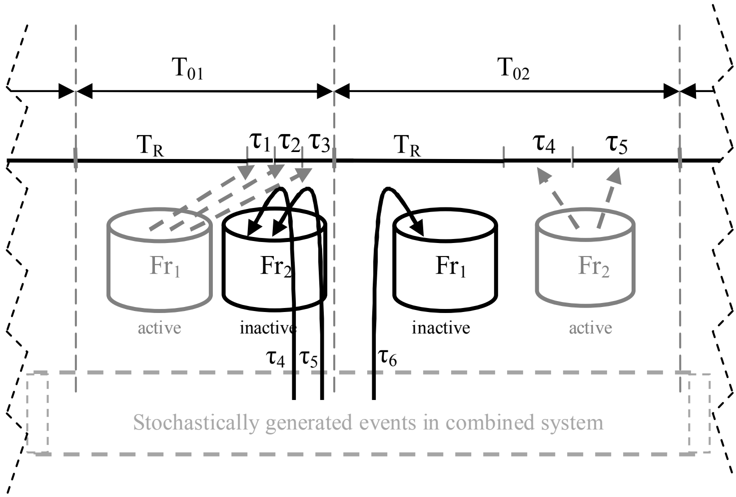


***Fig. 25** Executing service program blocks of events from active queue*

New events registered in the system are saved in the currently active queue of events and will be executed:

- either in the following cycle of sample period, when the inactive queue becomes an active one (Fig. 25),
- or in the present cycle of the sample period; provided that the service routines of all events listed in the active queue have been executed, and the time reserved for performing the event part of combined systems has not exceeded its double T_{Podhad}, the inactive queue in the system changes to an active one, and, if events were registered in the buffer (in present sample period), their related part (corresponding to the residual time of sample period) will be carried out. Queue activity in a sample period may change just once (Fig. 26).

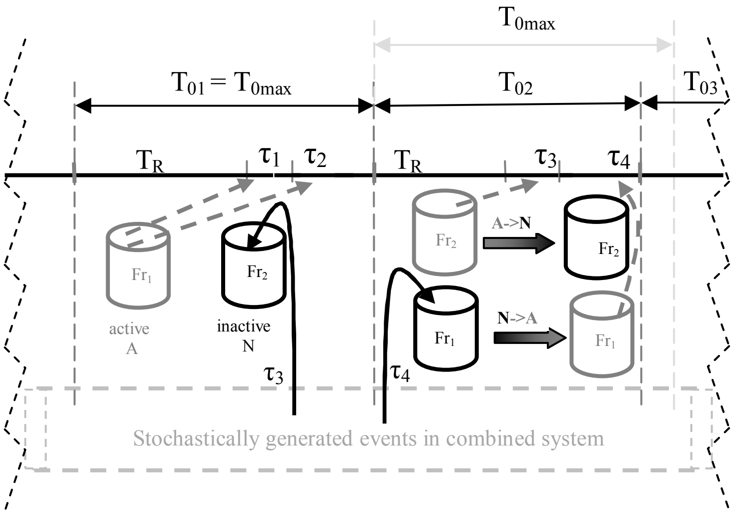


Fig. 26 *Change of queue activity in the second sample period*

While executing program blocks of events from the current active queue, the events registered in the system are arranged in the actually inactive queue of events according to the given planning algorithm. Once the change of queue activity has been performed in the course of one period, another attempt of re-activation in the same sample period will signal that all program blocks and instructions of all events generated in the combined dynamic system have been executed and arranged in both queues during the previous and current sample periods. The queue re-activation will terminate the current sample period at the time of the re-activation attempt, while starting the transition to the new cycle of the sample period by executing the operations related to the time-controlled part of a combined system within the reserved time T_R.

A high number of generated events may result in exceeding the time determined to execute the events in the system in the form of a probability estimate $2T_{Podhad}$. Within the proposed adaptive algorithm, there are two instances resulting in exceeding the maximum allowable event component of the sample period of combined systems:

There were no changes in the activity between the queues, and the system has generated a large number of events in the previous sample period with the implementation time of service programs of individual events, exceeding the possibilities of real time processing within the event component corresponding to the sample period of the programmable controller;

The activity queue is changed, i.e. the events in both queues are processed, and, in the past and current sample period, the system generated so many events within the time necessary for their operation, exceeding the real time processing capabilities in the limited time-space event component of the sample period of programmable regulator (Fig. 27).

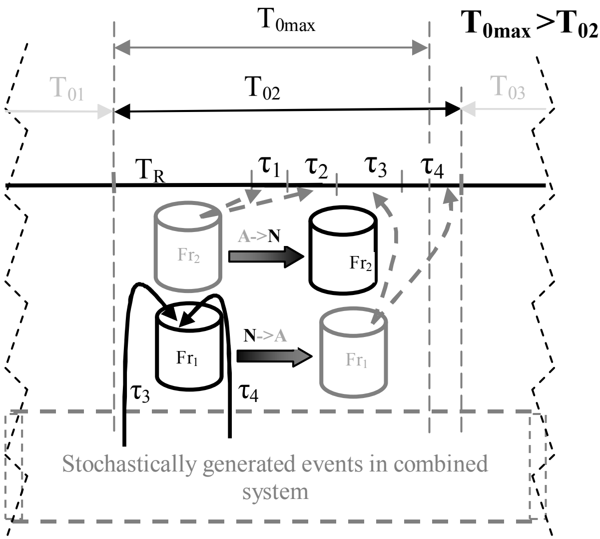


***Fig. 27** Exceeding the maximum value of time space reserved for the event component of sample period during the change of queue activity*

The adaptive part of the designed algorithm lies in re-determination of the probability estimate for the executing program and function blocks stochastically generated in a combined dynamic system. Re-determination of the event time component as well as the setup of a new sample period in a control element both take place if the maximum time space reserved for servicing the events generated in the system is exceeded consecutively N-times, while the value of variable N is defined in the declaration part of control algorithm. Unexecuted events of the active queue are assigned to an inactive queue so that, after its activation, all service routines of related events are executed in the following period (Fig. 28).

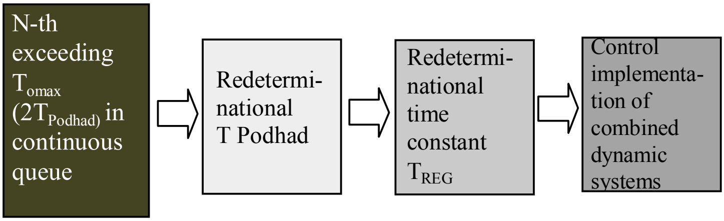


Fig. 28 *Redetermination in post-regulation part of algorithm*

3. 3 Impact of random component of sample period on combined dynamic system

Verification of the impact of the stochastic component on the combined dynamic system was conducted by means of a program designed and implemented in the environment of KDevelop language C++ under the Linux core of the Ubuntu system. The program performs conversion of differential equation coefficients of the regulation system in each step, based on the sample period consisting of the time necessary for the cyclic processing of instructions of the time-controlled system on one hand, and the time necessary for the implementation of the stochastic part of the system in the form of randomly generated events on the other hand. The continuous part of the regulation system is defined by the transfer of the regulated system in the form:

$$F_S(p) = \frac{5}{1+10p}, \tag{58}$$

and regulator transfer

$$F_R(p) = \frac{0{,}04}{p}. \tag{59}$$

A part of the regulatory circuit is a sampler with the sample period $T_0=5s$. Transfer of the disconnected regulation circuit is equal to

$$W(o) = \frac{5}{1+10p} \cdot \frac{0{,}04}{p}, \tag{60}$$

total transfer of the regulation circuit is then defined by the equation

$$W = \frac{W(o)}{1+W(o)} = \frac{\frac{5}{1+10p} \cdot \frac{0{,}04}{p}}{1+\frac{5}{1+10p} \cdot \frac{0{,}04}{p}}. \tag{61}$$

Based on the Bode diagram (Fig. 29) and transient characteristics (Fig. 30), we can conclude that the continuous control loop is stable and in regulation, and where the unit step is entering the system, it is controllable and stable.

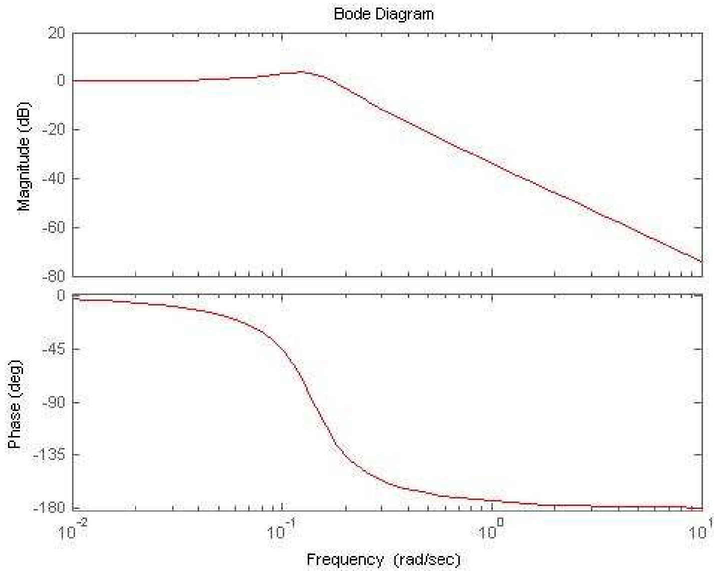


Fig. 29 *Bode diagram of regulation system*

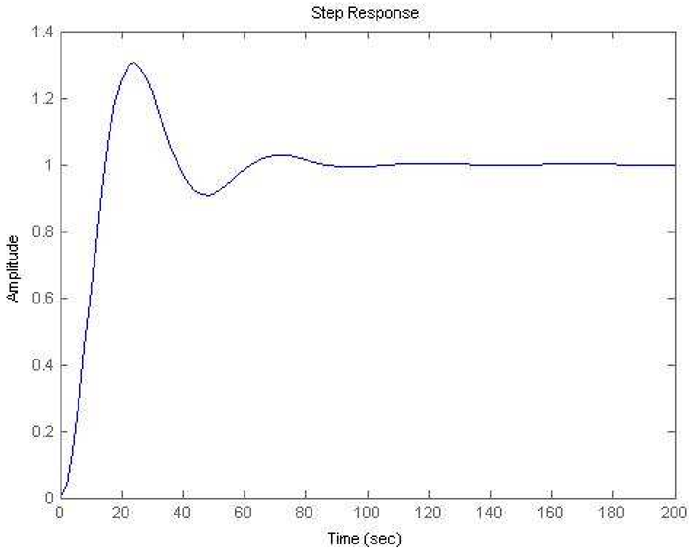


Fig. 30 *Step response of regulation system*

By means of differential equations for the regulated system

$$y(k)+a.y(k-1)=b.u(k), \tag{62}$$

the regulator

$$u(k)=r_{-1}.T.e(k)+u(k), \tag{63}$$

the differential member

$$e(k)=w(k)-y(k) \tag{64}$$

and their further modification, it is possible to establish a differential equation expressing the dependence of the controlling variable *w(k)* and the controlled variable *y(k)* in the form

$$y(k) - a.y(k-1) = b.r_{-1}.T_o.w(k-1) - b.r_{-1}.T_o.y(k-1) +$$
$$+ y(k-1) + a.y(k-2) \ , \tag{65}$$

where a = -0.606,

b = 1.97.

Transient characteristics of the discrete system, calculated by means of a differential equation at the unit step and without the influence of random component (T_o=5s), are illustrated in Fig. 31.

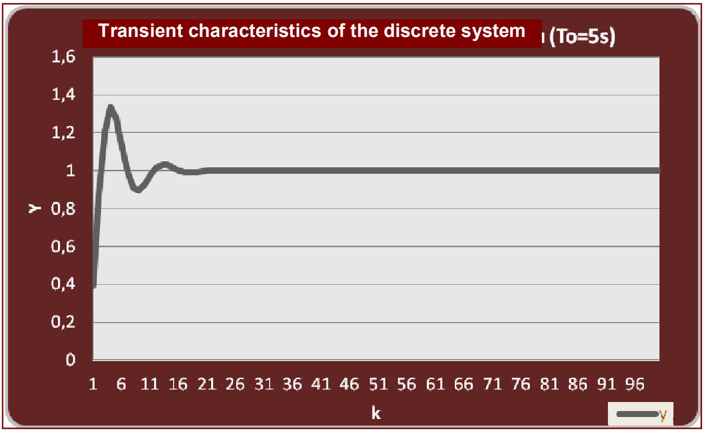


Fig. 31 *Transient characteristics of the discrete system without the influence of random constituent*

When assuming a stochastic constituent of randomly generated events in the system, the sample period acquires the form

$$T_o = T_R + T_P = T_R \ , \tag{66}$$

where ε represents the random component according to probability law closest to the event generation in the system. Calculation of T_p is just illustrative for the purposes of generating the model of combined system, and as an option for determining the stochastic part of the sample period. When considering a real combined system, it is necessary to define the mathematical description of the stochastic element behaviour corresponding to the given system as well as the generation of random events in it.

The combined discrete dynamic system selected in this way was tested for a unit step function in two models:

- the stochastic constituent ε of the sample period approaches normal probability distribution,
- the stochastic constituent ε of sample period approaches Poisson's probability distribution.

Each of the models was tested repeatedly (5-times) at various values describing related probability law. In the case of a model with normal probability distribution, calculation of random component ε with the following mean values and quadratic deviations (δ_x) were considered:

1. Mean value of random component at mean quadratic deviations:

   a) $\delta_{1a} = 1$,

   b) $\delta_{1b} = 0{,}5$.

2. Mean value of random component at mean quadratic deviations:

   a) $\delta_{2a} = 1$,

   b) $\delta_{2b} = 3$.

3. Mean value of random component at mean quadratic deviations:

a) $\delta_{3a} = 1$,

b) $\delta_{3b} = 3$,

c) $\delta_{3c} = 5$.

Calculation of random component ε for Poisson's probability distribution was conducted for the following mean values λ_x:

a) $\lambda_1=1$,

b) $\lambda_2=3$,

c) $\lambda_3=5$.

In order to monitor the model behaviour, the author of this monograph developed a program enabling acquisition of information regarding the status in each step of the discrete system, as well as recording results into the file in CSV format. Both file and results can be analysed by any tool supporting data-loading from the related file. In the case of the aforementioned models, the analysis and transient functions rendering were carried out in Excel. Special modules supporting the loading of randomly generated data into vectors were used to generate random numbers according to the relevant probability laws. Each group of values used to calculate random component ε was subjected to the simulation program 5-times. The result was transient characteristics of the system influenced by a random time component of the sample period. A summary chart of all transient characteristics of a particular group and selected transient characteristics of the group with a more detailed view of the transient curve will be given for each group of values. The program performs the algorithm expressed by means of the flowchart illustrated in Fig. 32. Source codes are listed in Appendix D.

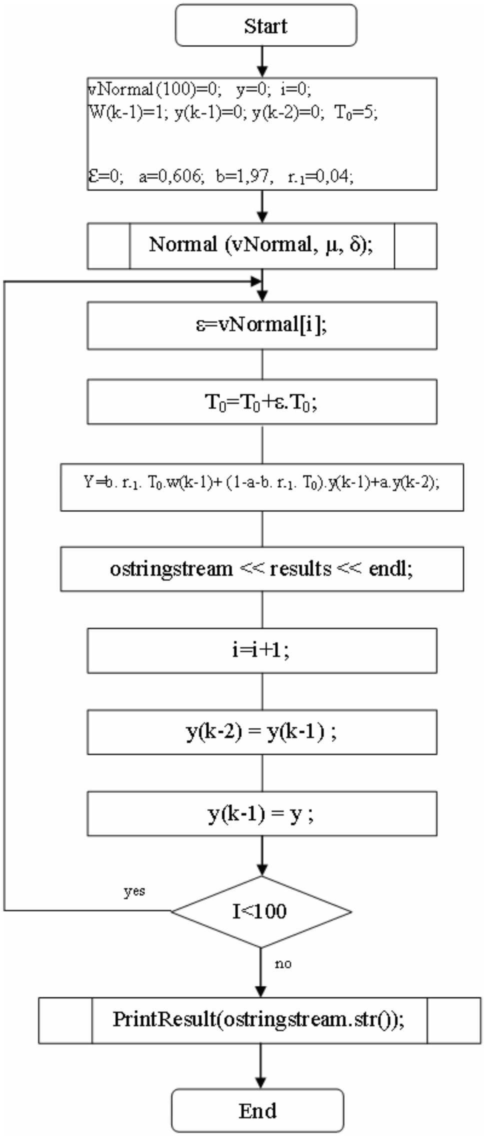


Fig. 32 *Algorithms for calculating the random component and differential equations for a discrete system*

3.3.1 Calculating the differential equation coefficients and the random component corresponding to normal probability distribution

The transient characteristics of a discrete system expressed by means of differential equations and tested 5-times consecutively by the program at the given values are illustrated in individual figures. Transient characteristics calculated from the average values of individual revisions are denoted in black colour. For the mean value, the random component of the sample period for the following mean quadratic deviations was gradually calculated:

a) $\delta_{3a} = 1$ – transient characteristics for individual program performance of differential equations of a discrete system are shown in Fig. 33. The graphs indicate that, at a given mean value and deviation, and owing to the random time constituent of the sample period, the system gradually stabilizes on the required value.

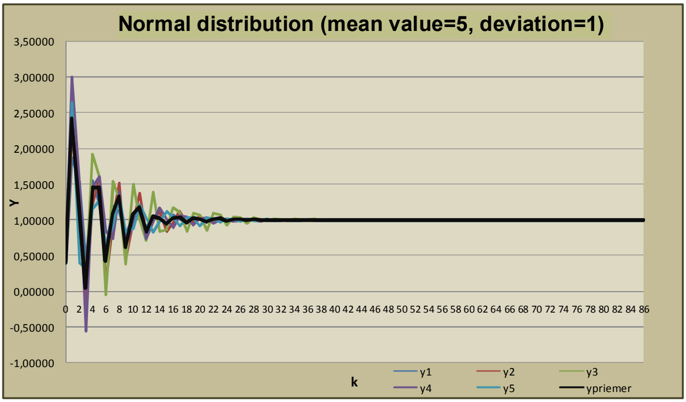


Fig. 33 *Transient characteristics at a* $\delta_{3a} = 1$

b) $\delta_{3b} = 3$ – Fig. 34. Unlike in the previous case, the increased influence of the random component on the overall system stability can be observed here. The increase of the mean quadratic deviation in the process of generating the value of the random component corresponding to normal probability distribution causes instability in the process regulation; as a result, the system starts oscillating.

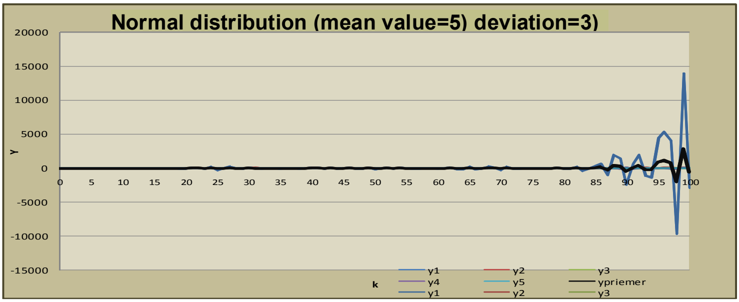


Fig. 34 *Transient characteristics at a* $\delta_{3b} = 3$

c) $\delta_{3c} = 5$ – Fig. 35. A high value of quadratic deviation causes the total instability of the system (its complete oscillation).

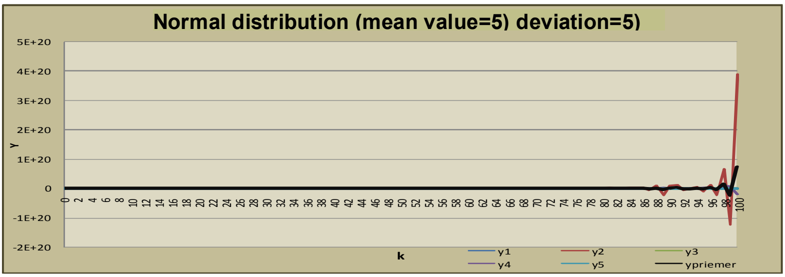


Fig. 35 *Transient characteristics at a* $\delta_{3c} = 5$

Transient characteristics at the mean value of random component of sample period were implemented with the following mean quadratic deviations:

a) $\delta_{2a} = 1$ - Fig. 36. The random component of the sample period (compared to the system without a random component) caused rather over-regulation than instability of the system in the process of its regulation to a unit step input.

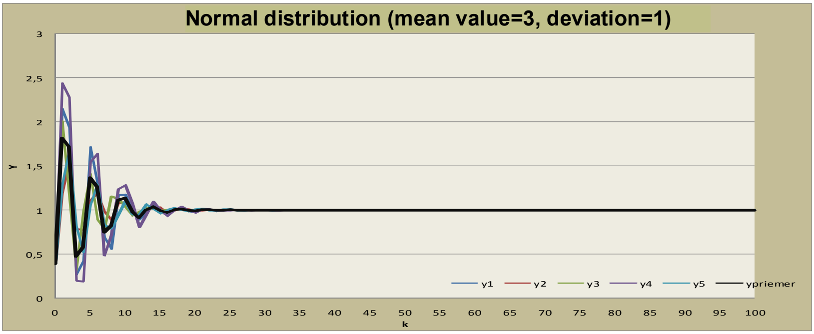


Fig. 36 *Transient characteristics at a* $\delta_{2a} = 1$

b) $\delta_{2b} = 3$ - Fig. 37. Owing to the high quadratic deviation, a random quadratic. Component in this case caused the system instability and its oscillation in the process of regulation to a unit step input.

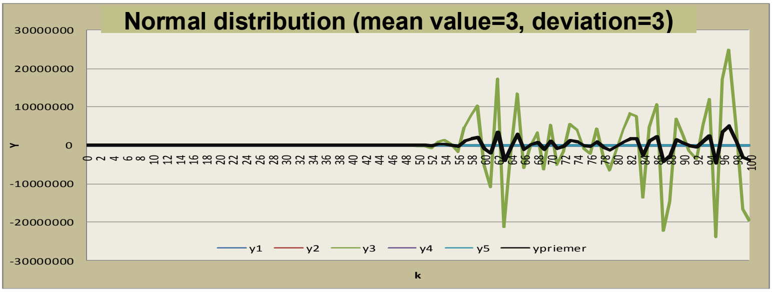


Fig. 37 Transient characteristics at a $\delta_{2b} = 3$

Final calculation of the random time component at normal probability distribution of event generation in a system concerned the transient characteristics at the mean value with the following mean quadratic deviations:

c) $\delta_{1a} = 1$ - Fig. 38. Transient characteristics indicate nearly trouble-free regulation and stabilisation of the whole system.

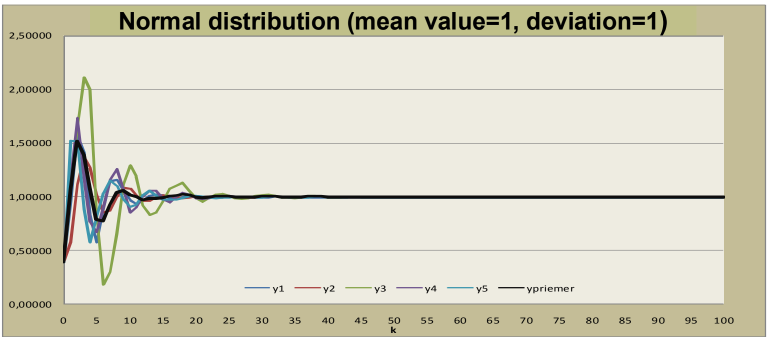


Fig. 38 Transient characteristics at $\delta_{1a} = 1$

d) $\delta_{1b} = 0{,}5$ - Fig. 39. Similarly to the previous case, system stabilisation after initial oscillation caused by the unit jump is evident.

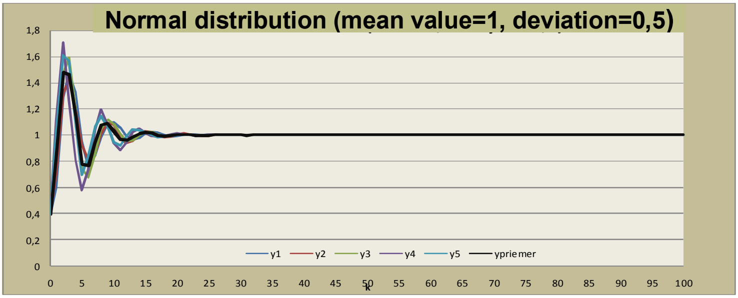


Fig. 39 *Transient characteristics at* $\delta_{1b} = 0{,}5$

The attained results allow us to conclude that the growing mean value and mean quadratic deviation influence the random time component of the sample period by increasing the system over-regulation and its inclination to instability or oscillation. Examples of calculations of differential equations of a discrete system and the system sample period of the system dependent on the random time constituents are listed in Appendix B.

3.3.2 Calculating the differential equation coefficients and the random component corresponding to Poisson's probability distribution

Coefficients of differential equation of a combined discrete system calculated for Poisson's time generator of random component ε were used to create transient characteristics for a discrete system. For each value λ_x, the program was repeated five times. The transient characteristics calculated from the mean values of individual repetitions are denoted by black color.

Transient characteristics for the mean value of stochastic events $\lambda_1=1$generation are shown in Fig. 40. The graph indicates that after the transient event had faded, and the system stabilized on the regulated value without the symptoms of instability or oscillations.

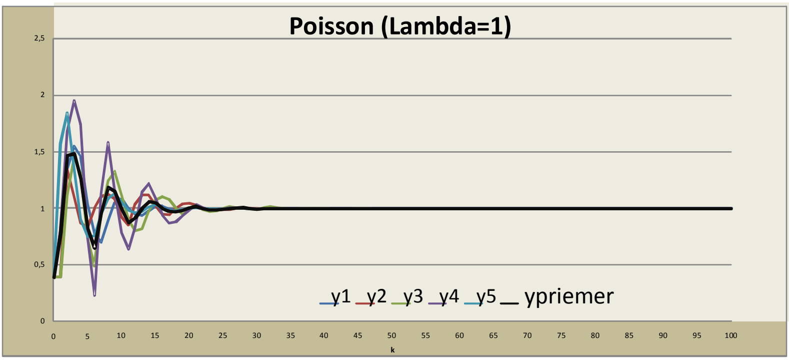


Fig. 40 *Transient characteristics for* $\lambda_1=1$

Similar characteristics were attained for the mean Poisson value $\lambda_2=3$. Transient characteristics are shown in Fig. 41. The only difference, when compared to the previous mean value, can be observed in the period of the transient event stabilisation in terms of the system responding to a unit step.

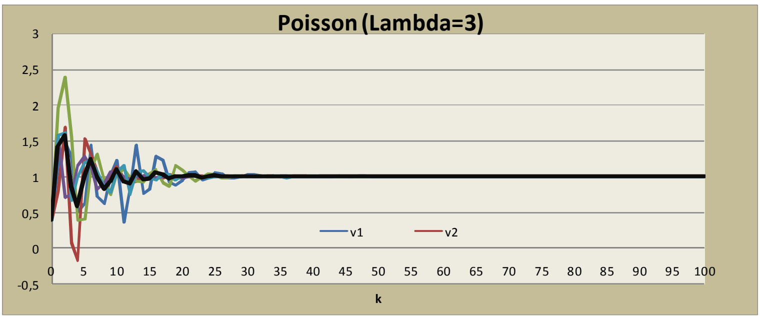


Fig. 41 *Transient characteristics for* $\lambda_2=3$

Final transient characteristics were drawn by using the results for mean value λ_3=5. The influence of the random component corresponding to the value of Poisson generator is shown in Fig. 42.

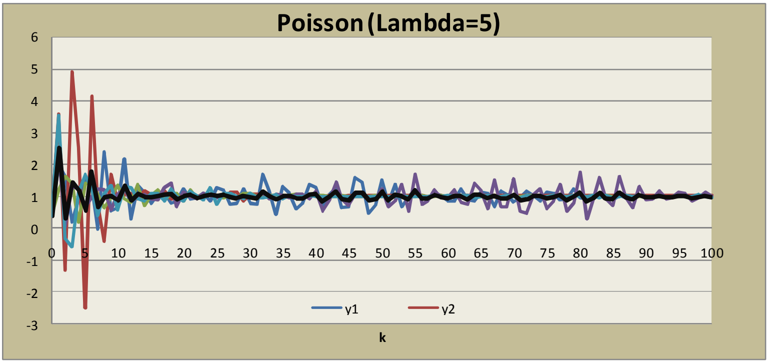


***Fig. 42** Transient characteristics for λ_3=5*

As the transient characteristics attained by the designed program suggest, the same outcome applies to both: the sample period depending on the random time component corresponding to Poisson's probability distribution on one hand, as well as the normal distribution on the other hand. The higher the mean Poisson's value influencing the sample period of the combined discrete system, the higher the system over-regulation and probability of its oscillation, hence the prolonged time of the system's stabilisation on the required value.

CONCLUSION

Combined discrete dynamic systems represent a real-time system processing both control instructions based on monitoring, collecting and evaluating the system information, as well as stochastic events initiated e.g. by alarm states, technical information on the state of hardware modules, etc. Current automation solutions enable the processing of unexpected events in particular control systems (e.g. Simatic stations). Combination of the time-and-event-driven systems enables the maximum utilisation of both types of control systems, their optimisation on one hand and minimisation of the related defects on the other hand. Combined dynamic systems represent a principally promising area in terms of controlling large and high-performance automation systems with a potential link to the higher-level information systems (MES, ERP).

The main contribution of the present monograph consists of:

- Classification of the combined dynamic systems, and the design of procedures for determining the sample period in the combined discrete dynamic systems for the stochastic events occurring in the system according to the normal and Poisson's probability distribution.
- Designed procedures for determining the properties of combined dynamic systems:
  - Intensity of the stochastic event by means of white noise,
  - Energy spectrum of combined dynamic systems.
- Proposal of adaptive algorithms of the combined dynamic systems' implementation; processing the events and auto-correction of the stochastic event time component of a sample period.

- Procedures for analyzing the impact of the sample period random component on the combined dynamic system by means of software tools, and the verification models for the system corresponding to both normal distribution and Poisson's probability distribution.

LIST OF AUTHOR'S PUBLICATIONS

[1] MICHAĽČONOK, G., STRÉMY, M. Estimation of dynamic characteristics of vizualization software. In *CO-MA-TECH 2003:* Bratislava: STU, 2003, pp. 689-694. ISBN 80-227-1949-8

[2] MICHAĽČONOK, G., STRÉMY, M. Simulácia riadiacich systémov založených na funkčných blokoch riadenia pohybu (Simulation of control systems based on function blocks of motion control). In *Proceedings of the 6th International Scientific-Technical Conference PROCESS CONTROL – ŘÍP 2004*. Pardubice: University of Pardubice, 2004, R224/1-8. ISBN 80-7194-662-1

[3] MICHAĽČONOK, G., STRÉMY, M. The Combined discrete systems. In *CO-MA-TECH 2004:* [CD-ROM]. Bratislava: STU v Bratislave, 2004. ISBN 80-227-2117-4

[4] MICHAĽČONOK, G., STRÉMY, M. Control system with combined interruption. In *CO-MA-TECH 2005.* Bratislava: STU, 2005, pp.1100-1103. ISBN 80-227-2286-3

[5] MICHAĽČONOK, G., STRÉMY, M. Energy spectrum of the combined systems. In *CO-MA-TECH 2006*. Bratislava: STU, 2006, pp. 1206-1211. ISBN 80-227-2472-6

[6] MICHAĽČONOK, G., STRÉMY, M. Systémy riadenia s náhodnou periódou vzorkovania. In *Proceedings the 7th International Scientific-Technical Conference PROCESS CONTROL – ŘÍP 2006*. Pardubice: Univerzita Pardubice, 2006, R088a-1/R088a-5. ISBN 80-7194-860-8

[7] MICHAĽČONOK, G., STRÉMY, M. Analyzing the control systems with random sample time. In *Proceedings the International doctoral seminar 2006*. Bratislava: STU , 2006, pp. 230-235. ISBN 80-227-2387-8

[8] KUNÍK, S., STRÉMY, M., MUDRONČÍK, D., KOPČEK, M. Virtual and analogue model of a continuous stirred tank reactor. In *Proceedings Process Control 2007 - 16th International Conference* [CD-ROM]. Bratislava: STU CHtF, 2007. ISBN 978-80-227-2677-1

[9] ELIÁŠ, A., STRÉMY, M. Návrh virtuálneho laboratória pre PLC (Design of virtual laboratory). In *Proceedings Applied Natural Sciences 2007.* Trnava: UCM, 2007, pp. 394-398. ISBN 978-80-89220-91-5

[10] MICHALÍK, F., STRÉMY, M. Grafické rozhranie virtuálneho regulátora (Graphical interface of virtual regulator) In *Proceedings*

Applied Natural Sciences 2007. Trnava: UCM, 2007, pp. 399-404. ISBN 978-80-89220-91-5

[11] KUDLA, M., STRÉMY, M. Alternatívne metódy ukladania pološtruktúrovaných dát (Alternative methods of storing semi-structured data). In *Proceedings of Applied Natural Sciences 2007.* Trnava: UCM, 2007, pp. 405-410. ISBN 978-80-89220-91-5

[12] STRÉMY, M, ELIÁŠ, A. et al. Virtual Laboratory Implementation. In *ECUMICT 2008 - Proceedings of the Third European Conference on the Use of Modern Information and Communication Technologies*. Gent: Nevelland v.z.w., 2008, pp. 385-393. ISBN 9-78908082-553-6

[13] ELIÁŠ, A., STRÉMY, M., KUDLA, M. Communication Interface for Virtual Devices. In *ECUMICT 2008 - Proceedings of the Third European Conference on the Use of Modern Information and Communication Technologies*. Gent: Nevelland v.z.w., 2008, pp. 133-138. ISBN 9-78908082-553-6

[14] STRÉMY, M., ELIÁŠ, A. Virtual logic module Logo!12/24RC. In *Proceedings the 8th International Scientific-Technical Conference PROCESS CONTROL – ŘÍP 2008*. Pardubice: University of Pardubice, 2008.

[15] ELIÁŠ, A., STRÉMY, M. Usage of Communication Dispatcher for Virtual. In *Proceedings the 8th International Scientific-Technical Conference PROCESS CONTROL – ŘÍP 2008*. Pardubice: University of Pardubice, 2008.

REFERENCES

[1] MUDRONČÍK, D. *Programové vybavenie riadiacich systémov (Program equipment of control systems)*. Bratislava: STU, 1994.

[2] MUDRONČÍK, D., ZOLOTOVÁ, I. *Priemyselné programovateľné regulátory (Industrial programmable regulators)*. Bratislava: Elfa, 2000. ISBN 80-88964-45-8

[3] SMUTNÝ, Ľ. Nové trendy v distribuovaných systémech řízení (New trends in distrideted control systems). In *Sborník ASŘ´97* [online]. Updated 1997-6-10 [cit. 2006-3-5]. Available on: <http://www.fs.vsb.cz/akce/1997/ASR97/ sbornik/sbornik.htm >.

[4] DORF R. C. *Modern Control Systems*. 5th ed. Davis (USA): Addison-Wesley Longman Publishing Co., 1989. ISBN 0-201-14278-3

[5] ČERVENKA, Z. Současné trendy v průmyslovém automatizačním softwaru pro průmyslovou automatizaci (Current trends in automation software for industrial automation). In *Automa* [online]. 2003, No. 3 [cit. 2006-2-2]. Available on: <http://www.automa.cz/automa/2000/au030046.htm>.

[6] HOLUB, G. Prognóza vývoje prodeje automatizační techniky (Prognosis of sales of automation technology). In *Automa* [online]. 2003, No. 4 [cit. 2006-5-20]. Available on: <http://www.odbornecasopisy.cz/automa/2003/au040342.htm>.

[7] TC1 – Standards [online]. [cit. 2004-11-15]. Available on: <http://www.plcopen.org/>.

[8] LEWIS, R. *Modelling control systems using IEC 61499*. London (United Kingdom): The Institution of Electrical Engineers, 2001. ISBN 0-85296-796-9

[9] SHINNERS, S. M. *Advanced Modern Control System Theory and Design*. USA: Wiley-Interscience, 1998. ISBN 0-471-31857-4

[10] ČEŠKA, M. *Petriho sítě (Petri nets)*. Brno: Akademické nakladatelství CERM, 1994. ISBN 80-85867-35-4

[11] HRÚZ, B. Discrete event systems modeling and real-time control. In *Journal of Electrical Engineering*, Vol. 45, No 10, pp. 363-370.

[12] HRÚZ, B., MRAFKO, L. *Modelovanie a riadenie diskrétnych udalostných dynamických systémov s využitím Petriho sietí a iných nástrojov (Modelling and control od discrete event dynamic systems utilising Petri nets and other tools)*. Bratislava: STU, 2003.

ISBN 80-227-1883-1

[13] DAVID, R., ALLA, H. *Petri nets and Grafcet*. New York (USA): Prentice Hall, 1992. ISBN 0-13-327537-X

[14] CARROLL, J., LONG, D. *Theory of finite automata with an introduction to formal languages*. Englewood Cliffs: Prentice-Hall, 1989. ISBN 0-13-913708-4

[15] MARTINS DE CARVALHO, J. L. *Dynamical systems and automatic control*. New York : Prentice-Hall, 1993, 444 s. ISBN 0-13-221755-4

CASSANDRAS, CHRISTOS G. *Discrete event systems: Modeling and performance analysis*. Boston (USA): Irwin, 1993. ISBN 0-256-11212-6

[16] LEWIS, R. *Programming industrial control systems using IEC 1131-3*. Revised edition. London (United Kingdom): The institution of Electrical Engineers, 1998. ISBN 0-85296-950-3

[17] FRANKLIN, G. F., POWELL, J. D., ABBAS EMAMI-NAEINI. *Feedback Control of Dynamic Systems*. Boston (USA): Addison-Wesley Longman Publishing Co., 1993. ISBN 0-201-52747-2

[18] FRANKLIN G. F., POWELL J. D., WORKMAN M. L. *Digital Control of Dynamic Systems.* Second Edition. Reading (USA): Addison-Wesley Longman Publishing Co., 1990. ISBN 0-201-51884-8

[19] БЕСЕКЕРСКИЙ, В. А., ПОПОВ, Е.П. *Теория систем автоматического регулирования (Theory of automatic regulation systems)*. Москва: <<Наука>> publishing house, 1975.

[20] ИЗЕРМАН, Р. *Цифровые системы управления (Numerical control systems)*. Москва : Мир,1984. 541 p. ISBN 0387502661

[21] MICHAELI, L. *Modelovanie analógovo číslicových rozhraní (Modelling analog-numerical interfaces)*. Košice: Mercury-Smékal, 2001. ISBN 80-968550-1-8

[22] *Technical Data: CPU 31xC and CPU 31x* [CD-ROM]. 2004, Edition 08 [cit. 2005-12-18]. Siemens. Ident. no.: A5E00105475-05.

[23] *Programming with STEP 7 V5.3* [CD-ROM]. 2004, Edition 01 [cit. 2005-12-03]. Siemens. Ident. No.: A5E00261405-01.

[24] BALÁTĚ, J. et al. *Technické prostředky automatického řízení (Technical means of automatic control)*. Praha: STNL, 1986. ISBN 04-538-86

[25] BANKS, J. et al. *Discrete-event system simulation*. New Jersey (USA): Prentice Hall, 2001. ISBN 0-13-088702-1

[26] ЛЕВИН, Б. Р. *Теория случайных процессов и ее применение в радиотехнике (Theory of random processes and its application in radiotechnology)*. Москва: Советское радио publishing house, 1960. 662 p.

[27] ONDRÁČEK, O. *Signály a sústavy (Signals and systems)*. Bratislava: STU, 2003. 341 p. ISBN 80-227-1875-0

[28] БЕСЕКЕРСКИЙ, B.A. *Сборник задач по теории автоматического регулирования и управления (Collection of tasks in theory of regulation and control)*. Москва: <<Наука>> publishing house, 1969. 588 p.

[29] VAN DER WAERDEN, B. L. *Einfache Herleaitung der Stirlingschen Formel (Simple expalnation of Stirling formula)*. [s.l.], 1936.

Appendix A – Overview of interruptions and organisation blocks

Type of interruption	Organisation block	Priority
Main scanning cyclic program	OB1	1
Interruptions in daytime	OB10 - OB17	2
Delayed interruptions	OB20 OB21 OB22 OB23	3 4 5 6
Cyclic interruptions	OB30 OB31 OB32 OB33 OB34 OB35 OB36 OB37 OB38	7 8 9 10 11 12 13 14 15
Hardware interruptions	OB40 OB41 OB42 OB43 OB44 OB45 OB46 OB47	16 17 18 19 20 21 22 23
Synchronic cyclic interruptions	OB61 OB62 OB63 OB64	25
Redundancy errors	OB70 OB72	25 28
Asynchronic errors	OB80 Timing error OB81 Supply error OB82 Diagnostic error	25

	OB84 Processor hardware error OB85 Program error OB87 Communication error	
Background cycle	OB90	29
Starting blocks	OB100 Restart OB101 Restart OB102 Cold restart	27 27 27
Synchronic errors	OB121 Program error OB122 Access error	

Appendix B – Samples of calculating random element in normal probability distribution ($\mu_{a1} = 1, \delta_{1a} = 1$)

y	Step	Calculation T_o	Differential equation - Y	Result
0.39400	0	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 0.394
0.96251	1	T= 5 + 1.44293* 5 = 12.2146	y=0.962513 w(k-1)+ 0.643487 y(k-1) +-0.606 y(k-2)	*** y= 0.962513
1.57925	2	T= 5 + 1.26497* 5 = 11.3248	y=0.892398 w(k-1)+ 0.713602 y(k-1) +-0.606 y(k-2)	*** y= 1.57925
1.40670	3	T= 5 + 1.39365* 5 = 11.9683	y=0.943099 w(k-1)+ 0.662901 y(k-1) +-0.606 y(k-2)	*** y= 1.4067
0.83199	4	T= 5 + 1.93404* 5 = 14.6702	y=1.15601 w(k-1)+ 0.449988 y(k-1) +-0.606 y(k-2)	*** y= 0.831986
0.57938	5	T= 5 + 0.445205* 5 = 7.22602	y=0.569411 w(k-1)+ 1.03659 y(k-1) +-0.606 y(k-2)	*** y= 0.579377
0.89539	6	T= 5 + 1.83055* 5 = 14.1527	y=1.11524 w(k-1)+ 0.490765 y(k-1) +-0.606 y(k-2)	*** y= 0.89539
1.14555	7	T= 5 + 0.423198* 5 = 7.11599	y=0.56074 w(k-1)+ 1.04526 y(k-1) +-0.606 y(k-2)	*** y= 1.14555
1.15871	8	T= 5 + 1.41411* 5 = 12.0706	y=0.951161 w(k-1)+ 0.654839 y(k-1) +-0.606 y(k-2)	*** y= 1.15871
1.05880	9	T= 5 + 0.725234* 5 = 8.62617	y=0.679742 w(k-1)+ 0.926258 y(k-1) +-0.606 y(k-2)	*** y= 1.0588
0.96695	10	T= 5 + 0.351414* 5	y=0.532457 w(k-1)+ 1.07354 y(k-1) +-0.606 y(k-	*** y= 0.966947

		= 6.75707	2)	
0.93649	11	T= 5 + 0.935085* 5 = 9.67542	y=0.762423 w(k-1)+ 0.843577 y(k-1) +-0.606 y(k-2)	*** y= 0.936485
0.96117	12	T= 5 + 0.723886* 5 = 8.61943	y=0.679211 w(k-1)+ 0.926789 y(k-1) +-0.606 y(k-2)	*** y= 0.961165
1.00986	13	T= 5 + 1.20525* 5 = 11.0263	y=0.86887 w(k-1)+ 0.73713 y(k-1) +-0.606 y(k-2)	*** y= 1.00986
1.02647	14	T= 5 + 2.32115* 5 = 16.6057	y=1.30853 w(k-1)+ 0.297467 y(k-1) +-0.606 y(k-2)	*** y= 1.02647
1.01500	15	T= 5 + 1.0646* 5 = 10.323	y=0.813452 w(k-1)+ 0.792548 y(k-1) +-0.606 y(k-2)	*** y= 1.015
0.99258	16	T= 5 + 1.61754* 5 = 13.0877	y=1.03131 w(k-1)+ 0.574691 y(k-1) +-0.606 y(k-2)	*** y= 0.992581
0.98307	17	T= 5 + 0.393024* 5 = 6.96512	y=0.548852 w(k-1)+ 1.05715 y(k-1) +-0.606 y(k-2)	*** y= 0.983067
0.99617	18	T= 5 + 1.82806* 5 = 14.1403	y=1.11426 w(k-1)+ 0.491743 y(k-1) +-0.606 y(k-2)	*** y= 0.996169
1.00708	19	T= 5 + 0.965259* 5 = 9.8263	y=0.774312 w(k-1)+ 0.831688 y(k-1) +-0.606 y(k-2)	*** y= 1.00708
1.00733	20	T= 5 + 1.28053* 5 = 11.4027	y=0.898531 w(k-1)+ 0.707469 y(k-1) +-0.606 y(k-2)	*** y= 1.00733
1.00227	21	T= 5 + 0.804585* 5 = 9.02293	y=0.711007 w(k-1)+ 0.894993 y(k-1) +-0.606 y(k-2)	*** y= 1.00227
0.99836	22	T= 5 + -0.0495886* 5	y=0.374462 w(k-1)+ 1.23154 y(k-1) +-0.606 y(k-	*** y= 0.998355

		= 4.75206	2)	
0.99812	23	T= 5 + 2.29869* 5 = 16.4934	y=1.29968 w(k-1)+ 0.306317 y(k-1) +-0.606 y(k-2)	*** y= 0.998121
0.99890	24	T= 5 + 0.242205* 5 = 6.21102	y=0.489429 w(k-1)+ 1.11657 y(k-1) +-0.606 y(k-2)	*** y= 0.998898
1.00026	25	T= 5 + 1.06201* 5 = 10.31	y=0.81243 w(k-1)+ 0.79357 y(k-1) +-0.606 y(k-2)	*** y= 1.00026
1.00088	26	T= 5 + 1.05264* 5 = 10.2632	y=0.808741 w(k-1)+ 0.797259 y(k-1) +-0.606 y(k-2)	*** y= 1.00088

Appendix C – Samples of calculating random element in Poisson's probability distribution ($\lambda_1 = 1$)

y	Step	Calculation T$_o$	Calculation Y	Result
0.394	0	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 0.394
0.788	1	T= 5 + 1* 5 = 10	y=0.788 w(k-1)+ 0.818 y(k-1) +-0.606 y(k-2)	*** y= 0.788
1.34906	2	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 1.34906
1.55153	3	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 1.55153
1.45692	4	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 1.45692
1.03954	5	T= 5 + 1* 5 = 10	y=0.788 w(k-1)+ 0.818 y(k-1) +-0.606 y(k-2)	*** y= 1.03954
0.771024	6	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 0.771024
0.698521	7	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 0.698521
0.89215	8	T= 5 + 1* 5 = 10	y=0.788 w(k-1)+ 0.818 y(k-1) +-0.606 y(k-2)	*** y= 0.89215
1.05198	9	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 1.05198
1.0874	10	T= 5 + 2* 5 = 15	y=1.182 w(k-1)+ 0.424 y(k-1) +-0.606 y(k-2)	*** y= 1.0874
1.00556	11	T= 5 + 2* 5 = 15	y=1.182 w(k-1)+ 0.424 y(k-1) +-0.606 y(k-2)	*** y= 1.00556
0.949393	12	T= 5 + 2* 5 = 15	y=1.182 w(k-1)+ 0.424 y(k-1) +-0.606 y(k-2)	*** y= 0.949393
0.935297	13	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 0.935297
0.977741	14	T= 5 + 1* 5 = 10	y=0.788 w(k-1)+ 0.818 y(k-1) +-0.606 y(k-2)	*** y= 0.977741
1.021	15	T= 5 + 1* 5 = 10	y=0.788 w(k-1)+ 0.818 y(k-1) +-0.606 y(k-2)	*** y= 1.021
1.02239	16	T= 5 + 2* 5 = 15	y=1.182 w(k-1)+ 0.424 y(k-1) +-0.606 y(k-2)	*** y= 1.02239
0.996768	17	T= 5 + 2* 5 = 15	y=1.182 w(k-1)+ 0.424 y(k-1) +-0.606 y(k-2)	*** y= 0.996768

0.983785	18	T= 5 + 1* 5 = 10	y=0.788 w(k-1)+ 0.818 y(k-1) +-0.606 y(k-2)	*** y= 0.983785
0.988695	19	T= 5 + 1* 5 = 10	y=0.788 w(k-1)+ 0.818 y(k-1) +-0.606 y(k-2)	*** y= 0.988695
1.00058	20	T= 5 + 1* 5 = 10	y=0.788 w(k-1)+ 0.818 y(k-1) +-0.606 y(k-2)	*** y= 1.00058
1.00732	21	T= 5 + 1* 5 = 10	y=0.788 w(k-1)+ 0.818 y(k-1) +-0.606 y(k-2)	*** y= 1.00732
0.999869	22	T= 5 + 3* 5 = 20	y=1.576 w(k-1)+ 0.03 y(k-1) +-0.606 y(k-2)	*** y= 0.999869
0.995506	23	T= 5 + 2* 5 = 15	y=1.182 w(k-1)+ 0.424 y(k-1) +-0.606 y(k-2)	*** y= 0.995506
0.994633	24	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 0.994633
0.996218	25	T= 5 + 0* 5 = 5	y=0.394 w(k-1)+ 1.212 y(k-1) +-0.606 y(k-2)	*** y= 0.996218
1.00314	26	T= 5 + 3* 5 = 20	y=1.576 w(k-1)+ 0.03 y(k-1) +-0.606 y(k-2)	*** y= 1.00314

Appendix D – Source program codes

```
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <cstdlib>
#include <time.h>
#include <iostream>
#include <sstream>
#include <fstream>
#include <queue>
#include "rng.h"

using namespace std;

bool PrintResult( std::string str )
{
            int i;
            string fname;
            fname="/home/mastre/vysledky.txt";

            std::cout << "prijate:" << str.c_str() << std::endl;

  ofstream cfile( fname.c_str() );
            if ( !cfile.is_open() )
            {
                                    std::cout << "Can't open file '" << fname.c_str()
<< "'" << std::endl;
                                    return false;
            }

                        cfile << str.c_str() << std::endl;

                        cfile.close();


            return true;
}
```

```
int genrand(int min, int max)
{
return min+(rand()%(++max-min));
}

int main(int argc, char *argv[])
{
int t;
float vysledok;
//y(k)-1,212.y(k-1)+0,606.y(k-2)=0,394.w(k-1)
//y(k)=0,394w(k-1)+1,212y(k-1) -0,606y(k-2)
 RNG rngRandom;

//***** NASTAVOVANIA ******//
vector<int> vPoisson(100);
vector<double>vNormal(100);
double stred=1, odchylka=0.5;
const string sDelim=";";
float w1=1,y1=0,y2=0,y=0,T=5, epsilon=0;
float a=-0.606, b=1.97, r=0.04;

rngRandom.poisson( vPoisson, stred); //lambda
rngRandom.normal( vNormal, stred, odchylka); //stredna hodnota, odchylka
std::ostringstream ostr;
ostr.str("");

for (unsigned int i = 0; i < vPoisson.size(); ++i)
   cout << vPoisson[i] << endl;

for (unsigned int i = 0; i < vNormal.size(); ++i)
   cout << vNormal[i] << endl;

cout<< "Normalove - stredna hodnota= "<<stred<< " rozptyl= "<<
odchylka<<endl;
ostr<< "Normalove - stredna hodnota= "<<stred<< " rozptyl= "<<
odchylka<<endl;

cout<< "y"<<sDelim<< "T"<<sDelim<<"krok"<< std::endl;
ostr<< "y"<<sDelim<< "T"<<sDelim<<"krok"<< std::endl;
```

```
cout<<  "----------------------     ZACIATOK   ------------------------------"<<
endl;

T=5; //perioda vzorkovania
y= b*r*T*w1 +(1-a-b*r*T)*y1+a*y2; //rovnica

ostr<< y << sDelim << T << sDelim <<"0"<< sDelim;

cout<< "T=" << T<< " +  0 * "<<T<< " = "<< T+0*T <<endl;
cout<< "y=" << b*r*T<< " w(k-1)+ " << (1-a-b*r*T) << " y(k-1) +"<< a<<
" y(k-2) ; *** y= "<< y <<endl;

ostr<< "T= 5 + " << "0* 5"<< " = "<< 5+0*5 <<sDelim;
ostr<< "y=" << b*r*T<< " w(k-1)+ " << (1-a-b*r*T) << " y(k-1) +"<< a<<
" y(k-2) ; *** y= "<< y <<sDelim;
ostr << std::endl;

//vector<double> v(10);
//x.normal( v, 5, 1); //stredna hodnota, rozptyl

for (unsigned int i = 0; i < vNormal.size(); ++i)
{
epsilon=(float)vNormal[i];
T=5;
T=5+epsilon*T;
y= b*r*T*w1 +(1-a-b*r*T)*y1+a*y2;
ostr<< y << sDelim << T << sDelim<< i+1<<sDelim;
cout<< "--------------------------  krok: "<< i+1 <<  "------------------"<< endl;
cout<< "T= 5 + " << epsilon << "* 5"<< " = "<< 5+epsilon*5 <<endl;
cout<< "y=" << b*r*T<< " w(k-1)+ " << (1-a-b*r*T) << " y(k-1) +"<< a<<
" y(k-2) ; *** y= "<< y <<endl;
cout<< endl;
ostr<< "T= 5 + " << epsilon << "* 5"<< " = "<< 5+epsilon*5 <<sDelim;
ostr<< "y=" << b*r*T<< " w(k-1)+ " << (1-a-b*r*T) << " y(k-1) +"<< a<<
" y(k-2) ; *** y= "<< y <<sDelim;
ostr << std::endl;
y2=y1;
y1=y;
```

```
w1=1;
}

w1=1; y1=0; y2=0; y=0; T=5; epsilon=0;
a=-0.606; b=1.97; r=0.04;
ostr<<                                    endl                                    <<
"*******************************************"<<endl;

cout<< "Poisson - lambda= "<<stred<<endl;
ostr<< "Poisson - lambda= "<<stred<<endl;

cout<< "y"<<sDelim<< "T"<<sDelim<<"krok"<< std::endl;
ostr<< "y"<<sDelim<< "T"<<sDelim<<"krok"<< std::endl;

cout<<   "----------------------     ZACIATOK   -------------------------------"<<
endl;

T=5; //perioda vzorkovania
y= b*r*T*w1 +(1-a-b*r*T)*y1+a*y2; //rovnica

ostr<< y << sDelim << T << sDelim <<"0"<< sDelim;

cout<< "T=" << T<< " +  0 * "<<T<< " = "<< T+0*T <<endl;
cout<< "y=" << b*r*T<< " w(k-1)+ " << (1-a-b*r*T) << " y(k-1) +"<< a<<
" y(k-2) ; *** y= "<< y <<endl;

ostr<< "T= 5 + " << "0* 5"<< " = "<< 5+0*5 <<sDelim;
ostr<< "y=" << b*r*T<< " w(k-1)+ " << (1-a-b*r*T) << " y(k-1) +"<< a<<
" y(k-2) ; *** y= "<< y <<sDelim;
ostr << std::endl;
for (unsigned int i = 0; i < vPoisson.size(); ++i)
{
epsilon=(float)vPoisson[i];
//y2=(float)v[i];
//y=0.394*w1+1.212*y1-0.606*y2;
T=5;
//y= b*r*T*w1 +y1-a*y1-b*r*T*y1+a*y2;
T=5+epsilon*T;
y= b*r*T*w1 +(1-a-b*r*T)*y1+a*y2;\
```

```
ostr<< y << sDelim << T << sDelim<< i+1<<sDelim;

cout<< "--------------------------  krok: "<< i+1 <<  "------------------"<< endl;
cout<< "T= 5 + " << epsilon << "* 5"<< " = "<< 5+epsilon*5 <<endl;
cout<< "y=" << b*r*T<< " w(k-1)+ " << (1-a-b*r*T) << " y(k-1) +"<< a<<
" y(k-2) ; *** y= "<< y <<endl;
cout<< endl;

ostr<< "T= 5 + " << epsilon << "* 5"<< " = "<< 5+epsilon*5 <<sDelim;
ostr<< "y=" << b*r*T<< " w(k-1)+ " << (1-a-b*r*T) << " y(k-1) +"<< a<<
" y(k-2) ; *** y= "<< y <<sDelim;
ostr << std::endl;

y2=y1;
y1=y;
w1=1;
}

if (!PrintResult(ostr.str())) cout<< "CHYBA PRI ZAPISE "<<endl;
return EXIT_SUCCESS;
}
```

CONTENTS